

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

17. 3. 2004

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日      2 0 0 3 年   3 月 1 9 日  
Date of Application:

出 願 番 号      特 願 2 0 0 3 - 0 7 6 0 7 1  
Application Number:  
[ST. 10/C]:      [ J P 2 0 0 3 - 0 7 6 0 7 1 ]

出 願 人      セイコーエプソン株式会社  
Applicant(s):

REC'D 29 APR 2004

WIPO

PCT

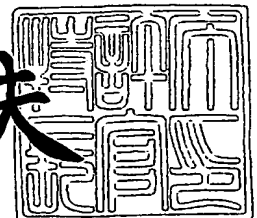
BEST AVAILABLE COPY

**PRIORITY DOCUMENT**  
SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH  
RULE 17.1(a) OR (b)

2 0 0 4 年   4 月 1 5 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 PA04F397

【提出日】 平成15年 3月19日

【あて先】 特許庁長官 太田 信一郎 殿

【国際特許分類】 H04N 1/00

【発明者】

    【住所又は居所】 長野県諏訪市大和三丁目3番5号 セイコーエプソン株式会社社内

    【氏名】 角谷 繁明

【特許出願人】

    【識別番号】 000002369

    【氏名又は名称】 セイコーエプソン株式会社

【代理人】

    【識別番号】 110000028

    【氏名又は名称】 特許業務法人 明成国際特許事務所

    【代表者】 下出 隆史

    【電話番号】 052-218-5061

【手数料の表示】

    【予納台帳番号】 133917

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 0105458

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ブロック内のエッジを検出しながら処理を行う画像処理装置、  
および画像処理方法

【特許請求の範囲】

【請求項 1】 第 1 の表色系により表現されたカラー画像データを、第 2 の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理装置であって、

前記第 1 の表色系により表現されたカラー画像データを前記第 2 の表色系によるカラー画像データに色変換する色変換手段と、

互いに隣接する画素を所定の複数個ずつ第 1 種のブロックとしてまとめて、該第 1 種のブロック内でのエッジの有無を検出するエッジ検出手段と、

前記第 1 種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する圧縮データ記憶手段と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第 2 種のブロックとしてまとめる圧縮データ解凍手段と、

前記第 2 種のブロックとしてまとめられたカラー画像データに基づいて、該ブロックに含まれる各画素についてのドットの形成有無を、前記第 2 の表色系の各色毎に該ブロック単位で判断するドット形成有無判断手段と

を備える画像処理装置。

【請求項 2】 請求項 1 記載の画像処理装置であって、

前記ドット形成有無判断手段は、前記第 2 種のブロックを構成する複数の画素の中で、エッジが検出されることなく圧縮されていた複数の画素については、該画素のカラー画像データを均一な画像データとして扱って、前記ドットの形成有無を判断する手段である画像処理装置。

【請求項 3】 請求項 1 記載の画像処理装置であって、

前記エッジ検出手段は、前記第 1 種のブロック内でのエッジの検出を、該ブロック内の各画素についての前記色変換後のカラー画像データに基づいて行う手段

であり、

前記圧縮データ記憶手段は、前記色変換後のカラー画像データを、前記第1種のブロック内でのエッジの有無に応じて圧縮した後、圧縮されたデータを前記メモリに記憶する手段である画像処理装置。

【請求項4】 請求項1記載の画像処理装置であって、

前記エッジ検出手段は、前記第1種のブロック内でのエッジの検出を、該ブロック内の各画素についての前記色変換前のカラー画像データに基づいて行う手段であり、

前記圧縮データ記憶手段は、前記エッジの検出されなかった前記第1種のブロックについて、前記色変換前のカラー画像データを圧縮した後、該圧縮したカラー画像データを前記第2の表色系のカラー画像データに色変換して前記メモリに記憶する手段である画像処理装置。

【請求項5】 請求項4記載の画像処理装置であって、

前記エッジ検出手段は、前記色変換前のカラー画像データに基づいてエッジが検出された前記第1種のブロックについては、前記色変換後のカラー画像データに基づいて再びエッジの検出を行う手段であり、

前記圧縮データ記憶手段は、前記色変換後のカラー画像データに基づいて再びエッジの有無を検出した前記第1種のブロックについては、該エッジの検出結果に応じて該色変換後のカラー画像データを圧縮した後、前記メモリに記憶する手段である画像処理装置。

【請求項6】 請求項1ないし請求項5のいずれかに記載の画像処理装置であって、

前記圧縮データ記憶手段は、エッジを含まない前記第1種のブロックについては、該ブロックに含まれる複数個分の画素の画像データを1画素分の画像データに圧縮して記憶する手段である画像処理装置。

【請求項7】 請求項6記載の画像処理装置であって、

前記圧縮データ記憶手段は、エッジを含まない前記第1種のブロックについては、該ブロックに含まれる複数個分の画素の画像データを、該画像データの平均値に圧縮して記憶する手段である画像処理装置。

【請求項 8】 請求項 6 記載の画像処理装置であって、

前記圧縮データ記憶手段は、エッジを含まない前記第 1 種のブロックについては、該ブロックに含まれる複数個分の画素の画像データを、該ブロック内から選択された 1 の画素の画像データに圧縮して記憶する手段である画像処理装置。

【請求項 9】 請求項 1 ないし請求項 5 のいずれかに記載の画像処理装置であって、

前記圧縮データ記憶手段は、前記第 1 種のブロック内にエッジが検出された場合には、該ブロックをエッジを含まないサブブロックに分割し、該サブブロックに含まれる画素について前記色変換後の画像データを圧縮した後、前記メモリに記憶する手段である画像処理装置。

【請求項 10】 請求項 1 ないし請求項 5 のいずれかに記載の画像処理装置であって、

前記圧縮データ記憶手段は、前記エッジの有無に応じて圧縮された色変換後のカラー画像データとともに、該カラー画像データが圧縮されている状態を表す圧縮情報を前記メモリに記憶する手段である画像処理装置。

【請求項 11】 請求項 1 ないし請求項 5 のいずれかに記載の画像処理装置であって、

前記圧縮データ解凍手段は、前記第 2 種のブロックとして、前記第 1 種のブロックと同一のブロックにまとめる手段である画像処理装置。

【請求項 12】 請求項 11 記載の画像処理装置であって、

前記エッジ検出手段および前記圧縮データ解凍手段は、2 行 2 列に配列された 4 つの画素を、それぞれ前記第 1 種のブロックおよび前記第 2 種のブロックとしてまとめる手段である画像処理装置。

【請求項 13】 第 1 の表色系により表現されたカラー画像データを、第 2 の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理装置であって、

前記第 1 の表色系により表現されたカラー画像データを前記第 2 の表色系によるカラー画像データに変換する色変換手段と、

互いに隣接する画素を所定の複数個ずつ第 1 種のブロックとしてまとめて、該

第1種のブロック内でのエッジの有無を検出するエッジ検出手段と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する圧縮データ記憶手段と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する圧縮画像データ解凍手段と、

前記解凍されたカラー画像データに基づいて、前記第2の表色系の各色毎にドットの形成有無を判断するドット形成有無判断手段と  
を備える画像処理装置。

【請求項14】 第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理方法であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する工程と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する工程と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する工程と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第2種のブロックとしてまとめる工程と、

前記第2種のブロックとしてまとめられたカラー画像データに基づいて、該ブロックに含まれる各画素についてのドットの形成有無を、前記第2の表色系の各色毎に該ブロック単位で判断する工程と

を備える画像処理方法。

【請求項15】 第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理方法であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する工程と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する工程と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する工程と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する工程と、

前記解凍されたカラー画像データに基づいて、前記第2の表色系の各色毎にドットの形成有無を判断する工程と

を備える画像処理方法。

【請求項16】 第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する方法を、コンピュータを用いて実現するためのプログラムであって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する機能と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する機能と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する機能と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第2種のブロックとしてまとめる機能と、

前記第2種のブロックとしてまとめられたカラー画像データに基づいて、該ブロックに含まれる各画素についてのドットの形成有無を、前記第2の表色系の各色毎に該ブロック単位で判断する機能と

を実現するプログラム。

【請求項 17】 第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する方法を、コンピュータを用いて実現するためのプログラムであって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する機能と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する機能と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する機能と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する機能と、

前記解凍されたカラー画像データに基づいて、前記第2の表色系の各色毎にドットの形成有無を判断する機能と

を実現するプログラム。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

この発明は、カラー画像データを変換する技術に関し、詳しくは、第1の表色系による画像データを第2の表色系の画像データに色変換した後、該色変換後のデータを、ドット形成の有無による表現形式の画像データに変換する技術に関する。

##### 【0002】

#### 【従来の技術】

インクジェットプリンタなど、印刷媒体上にドットを形成して画像を印刷する印刷装置は、コンピュータやデジタルカメラなどの画像機器の出力装置として広く使用されている。また近年では、液晶や有機LEDなどの画面上に輝点を形成



して画像を表示する画像表示装置も、こうした出力装置として使用され始めており、今後は、広く使用されるようになるものと考えられる。

#### 【0003】

コンピュータやデジタルカメラ上の画像を、これら出力装置を用いて表示するためには、画像データに、色変換処理およびハーフトーン処理と呼ばれる画像処理を施しておく必要がある。これは、画像データを作成する側（コンピュータなど）と画像データに基づいて画像を表示する側（印刷装置など）とで、カラー画像データの表現方法が以下のように異なっているためである。

#### 【0004】

先ず、コンピュータなどと印刷装置などとは、色相を表現するための方式が異なっている。すなわち、コンピュータなどでは、R（赤色）、G（緑色）、B（青色）を組み合わせる色相を表現するのに対して、印刷装置ではC（シアン色）、M（マゼンタ色）、Y（イエロ色）を基本とする組合せを用いて色相を表現する。このように、RGB各色を用いた色相の表現形式はRGB表色系と呼ばれることがある。これに対して、CMYの各色を基本として用いた色相の表現形式はCMY表色系と呼ばれることがある。また、液晶や有機LEDなど、コンピュータなどと同じくRGB表色系を採用している場合でも詳しく見れば、ほとんどの場合は、RGB各色の色目が微妙に異なっている。従って、同じRGB表色系を採用している場合でも、色目の違いを補正するために、カラー画像データの変換が必要となる。このように、コンピュータやデジタルカメラなどのカラー画像を正しい色相で表現するためには、コンピュータなどの側で採用されている表色系により表現されたカラー画像データを、印刷装置などの側で採用されている表色系によるカラー画像データに変換する必要がある。色変換処理とは、こうした変換を行う処理である。

#### 【0005】

また、コンピュータなどと印刷装置などとは、階調の表現方法も異なっている。すなわち、コンピュータなどでは、画像の最小単位である画素が、それ単独で多階調を表現可能であり、例えば1バイトの画像データであれば、個々の画素が単独で256階調を表現することが可能である。これに対して、印刷装置や画

像表示装置では、基本的には、ドットまたは輝点を形成するか否かの2つの状態しか取り得ず、従って、ドットや輝点の形成密度を変えることによって擬似的に階調を表現する方法が主に用いられる。このことから、画素毎に多階調を表現可能な画像データ（階調画像データ）を、擬似的に階調を表現した画像データ（擬似階調画像データ）に変換する処理が必要となる。ハーフトーン処理とは、こうした変換を行う処理である。

#### 【0006】

ハーフトーン処理の手法には種々の手法が提案されているが、代表的なものに「誤差拡散法」と呼ばれる手法がある。誤差拡散法は、着目画素にドットを形成したこと、あるいはドットを形成しなかったことによって生じる階調表現の誤差を、着目画素周辺の未判断画素に拡散して記憶しておき、未判断画素についてのドット形成有無を判断するにあたっては、周辺画素から拡散されてきた誤差を解消するように、ドットの形成有無を判断する手法である。誤差拡散法では、このように周辺画素で発生した階調表現の誤差を解消しながらハーフトーン処理を行うために、変換により発生する誤差を抑制して画像データを精度良く変換することが可能である。もっとも、誤差拡散法では、誤差を周辺の画素に拡散するたびに、メモリに対する読み出しおよび書き込みが発生するため、処理に時間がかかり、従って、画像の表示に時間がかかってしまう傾向がある。

#### 【0007】

こうした誤差拡散法の欠点を補いながら、カラー画像データに色変換処理と、誤差拡散法によるハーフトーン処理とを行う場合、それぞれに着眼点の異なった大きく3つの方式が考えられる。

#### 【0008】

先ず、第1の方式は、色変換処理と、誤差拡散法によるハーフトーン処理とを1画素毎に行う方式である。すなわち、処理する画素を1つ選択して、色変換処理とハーフトーン処理とを続けて行った後、ハーフトーン結果をメモリに記憶する。1つの画素についての処理が終了したら、また新たな画素を1つ選択して、色変換処理とハーフトーン処理とを行って、ハーフトーン結果をメモリに蓄えていく。こうした第1の方式によれば、色変換処理に続けてハーフトーン処理を行

っているので、色変換処理の結果は一時的なメモリ、すなわち中央演算装置（CPU）内のレジスタやキャッシュメモリなどの高速に読み書き可能なメモリに記憶することが可能となる。第1の方式は、一時的なメモリを活用することによって画像処理の高速化を図ろうとするものである。

#### 【0009】

第2の方式は、初めに色変換処理をまとめて行い、その後でハーフトーン処理を行う方式である。例えば、表示しようとしている画像の全画素について、あるいは画像中の1列分の画素について、初めに色変換処理を行って処理結果をメモリに記憶しておき、その後、色変換結果をメモリから読み出してハーフトーン処理を行う。第2の方式では、色変換処理だけを連続して行い、その後、ハーフトーン処理だけを連続して行う。このように同じ処理を連続して行うことにより、処理内容の簡素化あるいは並列化などの効果によって、画像処理の高速化を図ろうとするものである。

#### 【0010】

第3の方式は、第1の方式と第2の方式との中間的な方式であり、色変換処理とハーフトーン処理とを数画素単位で行う方式である（特許文献1）。例えば、縦横2列ずつの4つの画素を1組として、組単位で色変換処理とハーフトーン処理とを行う。このように、数画素単位で処理するのであれば、色変換処理の結果を、高速に読み書き可能な一時的なメモリに記憶しておくことが可能である。また、数画素分については同じ処理を続けて行うことになるので、この点では、第2の方式と同様な効果を狙ったものとも考えることもできる。

#### 【0011】

#### 【特許文献1】

特開 2002-185788 号公報

#### 【0012】

#### 【発明が解決しようとする課題】

しかし、これら方式は次のような課題を有しており、いずれも画像処理を十分に迅速化するには到っていないと言う問題があった。まず、第1の方式では、処理内容がめまぐるしく切り換わるために、ハーフトーン処理が複雑となり高速化

を図ることが困難である。すなわち、第1の方式では、色変換後、処理内容が直ちにハーフトーン処理に切り換わる。更にハーフトーン処理の内部でも、対象とする色成分が次々に切り換わることになる。結局、1画素毎にこうしてめまぐるしく処理を切替ながら画像処理を行うことになってしまうので、処理が複雑となり、高速化を図ることが困難となる。

#### 【0013】

第2の方式では、色変換処理をまとめて行い、ハーフトーン処理も色成分毎にまとめて行うことができるので、第1の方式のように、処理内容がめまぐるしく切り換わることはない。しかし、第2の方式では、色変換処理の結果を記憶しておくために大きなメモリ容量が必要となるので、色変換結果は、レジスタやキャッシュメモリなどのような一時的なメモリではなく、ハードディスクやRAMなどの通常のメモリ上に記憶されることになる。このため、第2の方式では、通常のメモリに対して多量のデータを読み書きしながら画像処理を行うことになり、そのために時間がかかってしまうので高速処理が困難となる。

#### 【0014】

更に、第3の方式では、数画素については、色成分毎にハーフトーン処理を行うことができるので、第1の方式に比べれば、処理内容が切り換わる頻度を抑制することができる。しかし、切り換えを抑制することによる効果は、数画素をまとめた程度では小さい上に、実際には、数画素ずつハーフトーン処理するために処理内容が複雑化してしまうので、十分な高速化を図ることは困難となっている。

#### 【0015】

この発明は、従来技術における上述の課題を解決するためになされたものであり、画像処理を複雑化させることなく、更なる迅速化が可能な技術を提供することを目的とする。

#### 【0016】

##### 【課題を解決するための手段およびその作用・効果】

上述の課題の少なくとも一部を解決するため、本発明の第1の画像処理装置は、次の構成を採用した。すなわち、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理装置であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する色変換手段と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出するエッジ検出手段と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する圧縮データ記憶手段と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第2種のブロックとしてまとめる圧縮データ解凍手段と、

前記第2種のブロックとしてまとめられたカラー画像データに基づいて、該ブロックに含まれる各画素についてのドットの形成有無を、前記第2の表色系の各色毎に該ブロック単位で判断するドット形成有無判断手段と

を備えることを要旨とする。

#### 【0017】

また、上記の画像処理装置に対応する本発明の第1の画像処理方法は、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理方法であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する工程と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する工程と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する工程と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第2種のブロックとしてまとめる工程と、

前記第2種のブロックとしてまとめられたカラー画像データに基づいて、該ブロックに含まれる各画素についてのドットの形成有無を、前記第2の表色系の各色毎に該ブロック単位で判断する工程と

を備えることを要旨とする。

#### 【0018】

かかる第1の画像処理装置および第1の画像処理方法においては、ドット形成有無の判断を、ブロック単位で行うことにより迅速に行うことができる。ドット形成有無を判断するための手法としては、誤差拡散法やディザ法などの周知の方法を適用することができる。こうしたドット形成の有無を判断するために用いられる色変換後の画像データは、圧縮された状態で一旦メモリに記憶されており、必要に応じてメモリから読み出しながらドット形成の有無が判断される。このため、ドット形成有無の判断をブロック単位で行うにも関わらず、処理が複雑化することがない。また、互いに隣接する画素を所定の複数個ずつ、第1種のブロックとしてまとめ、該ブロック内でのエッジの有無を検出した後、エッジの有無に応じて画像データを圧縮しているので、画像データの圧縮および解凍を迅速に行うことができる。このように、上記の画像処理装置および画像処理方法によれば、カラー画像データを色変換して、ドットの形成有無により表現されたデータに変換する処理を、処理の複雑化を伴うことなく、迅速に行うことが可能となる。

#### 【0019】

こうした第1の画像処理装置および画像処理方法においては、前記第2種のブロックを構成する各画素についてドット形成有無を判断するに際して、次のようにして判断しても良い。該第2種のブロックを構成する複数の画素の中で、第1種のブロックにエッジが含まれないとして圧縮されていた画素については、これら画素のカラー画像データを均一のデータとして扱って、ドット形成の有無を判断することとしても良い。ここで、均一な画像データとしては、これら複数の画素の中の1の画素についてのデータを用いても良いし、あるいはブロック内のデ

ータの平均値を用いても良い。

#### 【0020】

第2種のブロックを構成する画素の中、元々、エッジのないブロックに含まれているとして圧縮されていた画素については、画像データは近い値を取るものと考えられる。一方、ドット形成の有無を判断する際に、複数の画素の画像データを均一なデータとして扱うことができれば、ドット形成の有無を判断する処理の迅速化を図ることが可能である。従って、エッジが検出されないものとして圧縮されていた画素については、これら画素のデータを均一なデータとして扱うことで、画質に悪影響を与えることなく処理を迅速化することが可能となるので好適である。

#### 【0021】

また、画像処理装置および画像処理方法においては、前記第1種のブロックに含まれる各画素について、色変換後のカラー画像データに基づいてエッジの有無を検出し、該ブロック内でのエッジの有無に応じて、色変換後のカラー画像データを圧縮して前記メモリに記憶することとしても良い。

#### 【0022】

メモリに記憶されるカラー画像データは色変換後の画像データであるから、こうして色変換後の画像データに基づいてエッジを検出し、画像データを圧縮してやれば、第2の表色系を構成する各色毎に確実にエッジを検出して圧縮することが可能となるので好ましい。

#### 【0023】

あるいは、前記第1種のブロックに含まれる各画素について、色変換前のカラー画像データに基づいてエッジの有無を検出し、エッジの検出されなかったブロックについては、色変換前の画像データを圧縮することとしても良い。こうして、圧縮後のカラー画像データを色変換し、得られた画像データを前記メモリに記憶する。

#### 【0024】

こうすれば、エッジの検出されなかったブロックについては、該ブロックに含まれる画素をまとめて色変換することができるので、それだけ迅速に色変換を行

うことができ、延いては画像処理を迅速に行うことが可能となる。

#### 【0025】

更に、色変換前のカラー画像データに基づいてエッジの存在が検出された前記第1種のブロックについては、色変換後のカラー画像データに基づいて再びエッジの有無を各色毎に検出することとしても良い。そして、該エッジの検出結果に応じて、色変換後のカラー画像データを圧縮した後、該圧縮した画像データを前記メモリに記憶する。

#### 【0026】

こうすれば、色変換前に圧縮されなかったカラー画像データについても、色変換後の画像データに基づいて圧縮することができるので、画像データの圧縮率を向上させることができる。その結果、色変換後のカラー画像データを、前記メモリに対して迅速に読み書きすることが可能となり、延いては画像処理を高速化することができるので好ましい。

#### 【0027】

上述した第1の画像処理装置および画像処理方法においては、前記第1種のブロック内でエッジが検出されなかった場合は、該ブロック内に含まれる複数画素の画像データを、1画素分の画像データに圧縮して記憶することとしても良い。また、かかる圧縮に際しては、複数画素の中から1の画素を選択して、選択した画素の画像データに圧縮しても良いし、あるいは、複数画素についての画像データの平均値を算出して、複数画素分の画像データを得られた平均値に圧縮することとしても良い。

#### 【0028】

こうした方法によれば、複数画素の画像データを迅速に且つ効率よく圧縮することが可能であり、また、圧縮された画像データを迅速に解凍することが可能となるので好適である。

#### 【0029】

あるいは、上述した第1の画像処理装置および画像処理方法においては、前記第1種のブロック内にエッジの存在が検出された場合には、該ブロックをエッジを含まないサブブロックに分割し、該サブブロックに含まれる画素について前記



色変換後の画像データを圧縮した後、該圧縮された画像データを前記メモリに記憶することとしても良い。

#### 【0030】

こうして前記第1種のブロック内にエッジが検出された場合には、サブブロック単位でカラー画像データを圧縮してやれば、画像データの圧縮率を更に向上させることが可能となり、延いては、画像処理を高速化することが可能となるので好ましい。

#### 【0031】

更に、こうした各種の画像処理装置および画像処理方法においては、色変換後のカラー画像データを圧縮された状態で前記メモリに記憶するとともに、該画像データが圧縮されている状態を表す圧縮情報も、該メモリに記憶することとしても良い。

#### 【0032】

こうして圧縮された画像データとともに、圧縮情報を記憶しておけば、画像データを簡便に解凍することが可能となるので好ましい。

#### 【0033】

また、上述した第1の画像処理装置および画像処理方法においては、画像データを圧縮する際に用いられる前記第1種のブロックと、ドット形成有無を判断する際に用いられる前記第2種のブロックとを、同一のブロックとしても良い。

#### 【0034】

前記第1種のブロックと前記第2種のブロックとを同一のブロックとしておけば、圧縮されたカラー画像データを前記メモリから読み出して解凍する処理を簡素なものとすることができる。このため、画像処理を迅速に行うことが可能となるので好適である。

#### 【0035】

特に、前記第1種のブロックおよび前記第2種のブロックを、いずれも2行2列に配列された4つの画素から構成することとしても良い。

#### 【0036】

これらブロックに含まれる画素があまりに多くなると、エッジが含まれる確率

が高くなったり、画像処理が複雑となる。かといって、ブロックに含まれる画素があまりに少ないのでは、ブロック内のカラー画像データを圧縮しても、データ量がそれほど小さくならず、また、ドット形成有無の判断をブロック単位で行うことによるメリットも小さいので、画像処理を十分に迅速化することが困難となる。こうした観点から、2行2列に配列された4つの画素を第1種のブロックおよび第2種のブロックとしてまとめてやれば、画像処理を複雑化させることなく、処理速度を効果的に向上させることができる。

### 【0037】

また、上述した課題の少なくとも一部を解決するために、本発明の第2の画像処理装置は、次の構成を採用した。すなわち、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像処理装置であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する色変換手段と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出するエッジ検出手段と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する圧縮データ記憶手段と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する圧縮画像データ解凍手段と、

前記解凍されたカラー画像データに基づいて、前記第2の表色系の各色毎にドットの形成有無を判断するドット形成有無判断手段と

を備えることを要旨とする。

### 【0038】

また、上記の画像処理装置に対応する本発明の第2の画像処理方法は、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する画像

処理方法であって、

前記第 1 の表色系により表現されたカラー画像データを前記第 2 の表色系によるカラー画像データに色変換する工程と、

互いに隣接する画素を所定の複数個ずつ第 1 種のブロックとしてまとめて、該第 1 種のブロック内でのエッジの有無を検出する工程と、

前記第 1 種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する工程と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する工程と、

前記解凍されたカラー画像データに基づいて、前記第 2 の表色系の各色毎にドットの形成有無を判断する工程と

を備えることを要旨とする。

#### 【0039】

こうした第 2 の画像処理装置および画像処理方法においても、色変換後の画像データを圧縮された状態で一旦メモリに記憶しておく。そして、ドット形成の有無を判断する際には、必要に応じてメモリから読み出したデータを解凍しながら判断する。こうすれば、カラー画像データを色変換して、ドットの形成有無により表現されたデータに変換する処理を、処理の複雑化を伴うことなく、迅速に行うことが可能となる。

#### 【0040】

更に本発明は、上述した画像処理を実現するためのプログラムをコンピュータに読み込ませることにより、コンピュータを用いて実現することも可能である。従って、本発明は、次のようなプログラムあるいは該プログラムを記録した記録媒体として把握することもできる。すなわち、上述した第 1 の画像処理方法に対応する本発明の第 1 のプログラムは、

第 1 の表色系により表現されたカラー画像データを、第 2 の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する方法を、コンピュータを用いて実現するためのプログラムであって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する機能と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する機能と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する機能と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第2種のブロックとしてまとめる機能と、

前記第2種のブロックとしてまとめられたカラー画像データに誤差拡散法を適用して、該ブロックに含まれる各画素についてのドットの形成有無を、前記第2の表色系の各色毎に該ブロック単位で判断する機能と

を実現することを要旨とする。

#### 【0041】

また、上述した第1の画像処理方法に対応する本発明の第1の記録媒体は、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する方法を、コンピュータを用いて実現するためのプログラムを記録した記録媒体であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する機能と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する機能と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する機能と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍した後、互いに隣接する画素を所定の複数個ずつ第2種のブロックとして

まとめる機能と、

前記第2種のブロックとしてまとめられたカラー画像データに誤差拡散法を適用して、該ブロックに含まれる各画素についてのドットの形成有無を、前記第2の表色系の各色毎に該ブロック単位で判断する機能と

を、コンピュータで読み取り可能に記録したことを要旨とする。

#### 【0042】

更に、上述した第2の画像処理方法に対応する本発明の第2のプログラムは、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する方法を、コンピュータを用いて実現するためのプログラムであって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によるカラー画像データに色変換する機能と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する機能と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する機能と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する機能と、

前記解凍されたカラー画像データに基づいて、前記第2の表色系の各色毎にドットの形成有無を判断する機能と

を実現することを要旨とする。

#### 【0043】

また、上述した第2の画像処理方法に対応する本発明の第2の記録媒体は、

第1の表色系により表現されたカラー画像データを、第2の表色系を構成する各色についてのドットの形成有無によって表現された画像データに変換する方法を、コンピュータを用いて実現するためのプログラムを記録した記録媒体であって、

前記第1の表色系により表現されたカラー画像データを前記第2の表色系によ

るカラー画像データに色変換する機能と、

互いに隣接する画素を所定の複数個ずつ第1種のブロックとしてまとめて、該第1種のブロック内でのエッジの有無を検出する機能と、

前記第1種のブロックに含まれる各画素についての前記色変換後のカラー画像データを、該ブロック内でのエッジの有無に応じて圧縮された状態でメモリに記憶する機能と、

前記圧縮された状態で記憶されたカラー画像データを前記メモリから読み出して解凍する機能と、

前記解凍されたカラー画像データに基づいて、前記第2の表色系の各色毎にドットの形成有無を判断する機能と

を実現することを要旨とする。

#### 【0044】

こうしたプログラム、あるいは記録媒体に記憶されているプログラムをコンピュータに読み込ませ、該コンピュータを用いて上述の各種機能を実現すれば、画像処理を複雑化することなく、迅速にカラー画像データを変換することが可能となる。

#### 【0045】

##### 【発明の実施の形態】

本発明の作用・効果をより明確に説明するために、本発明の実施の形態を、次のような順序に従って以下に説明する。

A. 発明の概要：

B. 第1実施例：

B-1. 装置構成：

B-2. 画像処理の概要：

B-3. 第1実施例の色変換・圧縮処理：

B-4. ハーフトーン処理の概要：

C. 第2実施例：

D. 第3実施例：

E. 変形例：

## 【0046】

## A. 発明の概要:

実施例の詳細な説明を開始する前に、理解の便宜から、発明の概要について簡単に説明しておく。図1は、印刷システムを例にとって、本発明の概要を例示した説明図である。本印刷システムは、画像処理装置としてのコンピュータ10と、カラープリンタ20等から構成されている。コンピュータ10は、デジタルカメラやカラスキャナなどの画像機器からRGBカラー画像の階調画像データを受け取ると、該カラー画像データに所定の画像処理を施すことにより、カラープリンタ20で印刷可能な各色ドットの形成有無により表現された印刷データに変換する。かかる画像処理は、プリンタドライバ12と呼ばれる専用のプログラムを用いて行われる。尚、RGBカラー画像データは、各種アプリケーションプログラムを用いてコンピュータ10で作成することもできる。

## 【0047】

図1に例示のプリンタドライバ12は、色変換・圧縮モジュール、エッジ検出モジュール、ハーフトーンモジュールなどの複数のモジュールから構成されている。色変換・圧縮モジュールは、解像度が変換されたRGB画像データを受け取り、これをカラープリンタ20が形成可能なドットの色、すなわちシアン(C)色、マゼンタ(M)色、イエロ(Y)色の各色、あるいはこれに黒(K)色を加えた各色により表現されたCMY画像データに色変換する。色変換に先立って、画像データの解像度をカラープリンタ20が印刷するための解像度に変換しておくなど、必要に応じて前処理を施しておくこととしても良い。色変換後の画像データは、隣接する画素を所定の複数個ずつブロックとしてまとめて圧縮した後、このデータをメモリに記憶する。あるいは、後述するように色変換前に圧縮しておき、圧縮した状態で色変換することとしてもよい。画像データの圧縮に際しては、ブロック内にエッジが存在するか否かを検出し、この検出結果に応じて圧縮する。エッジの検出は、エッジ検出モジュールによって行われる。エッジの検出結果に応じた圧縮の詳細については、後述の各実施例において説明する。

## 【0048】

ハーフトーンモジュールは、圧縮して記憶されている画像データをメモリから

読み出して、隣接する画素を複数個ずつブロックとしてまとめた後、ブロック内に含まれる各画素についてドットの形成有無をブロック単位で行う。ここで、ハーフトーンモジュールが生成するブロックは、色変換・圧縮モジュールが生成するブロックと同一である必要はなく、例えば、ハーフトーンモジュールが生成するブロックと色変換・圧縮モジュールが生成するブロックとで、画素数を異ならせることも可能である。また、ドットの形成有無をブロック単位で行う方法についても、後述する方法を初めとして種々の方法を適用することができる。こうして、色変換によって生成した各色の画像データにハーフトーン処理を加えた後、得られたデータを印刷データとしてカラープリンタ 20 に出力する。カラープリンタ 20 への出力に先立って、プリンタがドットを形成する順番を考慮してデータの順番を並べ替えるなど、必要に応じて後処理を施すこととしてもよい。カラープリンタ 20 は、こうして得られた印刷データに基づいて、印刷媒体上に各色インクドットを形成することによってカラー画像を印刷する。

#### 【0049】

以上に説明した画像処理装置は、色変換後の画像データを圧縮してメモリに一旦記憶しておき、画像データを必要に応じてメモリから読み出して解凍してからハーフトーン処理を行う。ハーフトーン処理は複数画素ずつブロック単位で行っているため、迅速な処理が可能である。また、ハーフトーン処理では、必要に応じて画像データをメモリから読み出して処理しているため、ブロック単位で処理を行うにも関わらず、ハーフトーン処理を大幅に簡素化することができる。もちろん、画像データを圧縮したり解凍する処理や、画像データをメモリに対して読み書きする処理が新たに発生するが、後述するように、ブロック内でのエッジの有無に応じて圧縮することとすれば、極めて迅速に画像データを圧縮および解凍することが可能である。また、こうして圧縮された画像データを扱うことにより、メモリへの読み書きも極めて迅速に行うことができる。このため、本発明の画像処理によれば、処理内容を簡素化しつつ、画像処理全体としての処理速度を向上させることが可能となるのである。以下では、こうした画像処理の詳細について、実施例に基づき説明する。

#### 【0050】



## B. 第1実施例:

### B-1. 装置構成:

図2は、本実施例の画像処理装置としてのコンピュータ100の構成を示す説明図である。コンピュータ100は、CPU102を中心に、ROM104や、上述のメモリとして使用されるRAM106などを、バス116で互いに接続して構成された周知のコンピュータである。CPU102には、演算器やレジスタ、キャッシュなどが搭載されている。レジスタやキャッシュは、データを記憶する点ではRAM106と同じメモリの一種であるが、RAM106に比べてデータを遙かに高速に読み書きすることができる。

#### 【0051】

コンピュータ100には、フレキシブルディスク124やコンパクトディスク126のデータを読み込むためのディスクコントローラDDC109や、周辺機器とデータの授受を行うための周辺機器インターフェースP-I/F108、CRT114を駆動するためのビデオインターフェースV-I/F112等が接続されている。P-I/F108には、後述するカラープリンタ200や、ハードディスク118等が接続されている。また、デジタルカメラ120やカラースキャナ122等をP-I/F108に接続すれば、デジタルカメラ120やカラースキャナ122で取り込んだ画像を印刷することも可能である。また、ネットワークインターフェースカードNIC110を装着すれば、コンピュータ100を通信回線300に接続して、通信回線に接続された記憶装置310に記憶されているデータを取得することもできる。

#### 【0052】

図3は、第1実施例のカラープリンタ200の概略構成を示す説明図である。カラープリンタ200はシアン、マゼンタ、イエロ、ブラックの4色インクのドットを形成可能なインクジェットプリンタである。もちろん、これら4色のインクに加えて、染料濃度の低いシアン（淡シアン）インクと染料濃度の低いマゼンタ（淡マゼンタ）インクとを含めた合計6色のインクドットを形成可能なインクジェットプリンタを用いることもできる。尚、以下では場合によって、シアンインク、マゼンタインク、イエロインク、ブラックインク、淡シアンインク、淡マ

ゼンタインクのそれぞれを、Cインク、Mインク、Yインク、Kインク、LCインク、LMインクと略称することがあるものとする。

#### 【0053】

カラープリンタ200は、図示するように、キャリッジ240に搭載された印字ヘッド241を駆動してインクの吐出およびドットの形成を行う機構と、このキャリッジ240をキャリッジモータ230によってプラテン236の軸方向に往復動させる機構と、紙送りモータ235によって印刷用紙Pを搬送する機構と、ドットの形成やキャリッジ240の移動および印刷用紙の搬送を制御する制御回路260とから構成されている。

#### 【0054】

キャリッジ240には、Kインクを収納するインクカートリッジ242と、Cインク、Mインク、Yインクの各種インクを収納するインクカートリッジ243とが装着されている。キャリッジ240にインクカートリッジ242、243を装着すると、カートリッジ内の各インクは図示しない導入管を通じて、印字ヘッド241の下面に設けられた各色毎のインク吐出用ヘッド244ないし247に供給される。各色毎のインク吐出用ヘッド244ないし247には、48個のノズルNzが一定のノズルピッチkで配列されたノズル列が1組ずつ設けられている。

#### 【0055】

制御回路260は、CPU261と、ROM262、RAM263等から構成されている。制御回路260は、キャリッジモータ230と紙送りモータ235とを適切に駆動する。これにより、キャリッジ240の主走査と副走査とが制御される。また、制御回路260は、コンピュータ100から供給される印刷データに基づいて、各ノズルから適切なタイミングでインク滴を吐出する制御も司っている。こうして、制御回路260の制御の下、印刷媒体上の適切な位置に各色のインクドットを形成することによって、カラープリンタ200はカラー画像を印刷することができる。

#### 【0056】

尚、各色のインク吐出ヘッドからインク滴を吐出する方法には、種々の方法を

適用することができる。すなわち、 piezo素子を用いてインクを吐出する方式や、インク通路に配置したヒータでインク通路内に泡（バブル）を発生させてインク滴を吐出する方法などを用いることができる。また、インクを吐出する代わりに、熱転写などの現象を利用して印刷用紙上にインクドットを形成する方式や、静電気を利用して各色のトナー粉を印刷媒体上に付着させる方式のプリンタを使用することも可能である。

#### 【0057】

更には、吐出するインク滴の大きさを制御したり、あるいは一度に複数のインク滴を吐出して、吐出するインク滴の数を制御することにより、印刷用紙上に形成されるインクドットの大きさを制御可能な、いわゆるバリアブルドットプリンタを使用することもできる。

#### 【0058】

以上のようなハードウェア構成を有するカラープリンタ200は、キャリッジモータ230を駆動することによって、各色のインク吐出用ヘッド244ないし247を印刷用紙Pに対して主走査方向に移動させ、また紙送りモータ235を駆動することによって、印刷用紙Pを副走査方向に移動させる。制御回路260は、印刷データに従って、キャリッジ240の主走査および副走査を繰り返しながら、適切なタイミングでノズルを駆動してインク滴を吐出することによって、カラープリンタ200は印刷用紙上にカラー画像を印刷している。

#### 【0059】

#### B-2. 画像処理の概要:

図4は、本実施例の画像処理装置としてのコンピュータ100が、受け取った画像データに所定の画像処理を加えることによって、画像データを印刷データに変換する処理の流れを示すフローチャートである。かかる画像処理は、コンピュータ100のオペレーティングシステムがプリンタドライバ12を起動することによって開始される。以下、図4に従って、本実施例の画像処理について簡単に説明する。

#### 【0060】

プリンタドライバ12は、画像処理を開始すると、先ず初めに、変換すべきR

GBカラー画像データの読み込みを開始する（ステップS100）。次いで、取り込んだ画像データの解像度を、カラープリンタ200が印刷するための解像度に変換する（ステップS102）。カラー画像データの解像度が印刷解像度よりも低い場合は、線形補間を行うことで隣接する画像データ間に新たなデータを生成し、逆に印刷解像度よりも高い場合は一定の割合でデータを間引くことによって、画像データの解像度を印刷解像度に変換する。

#### 【0061】

こうして解像度を変換すると、カラー画像データの色変換および圧縮を行って、圧縮後の画像データをメモリすなわちRAM106に書き込む処理を行う（ステップS104）。色変換においては、R、G、Bの階調値の組み合わせによって表現されているカラー画像データを、C、M、Y、Kなどのカラープリンタ200で使用する各色の階調値の組み合わせによって表現された画像データに変換する処理を行う。色変換・圧縮処理の詳細については後述する。

#### 【0062】

プリンタドライバ12は、色変換・圧縮処理に続いてハーフトーン処理を開始する（ステップS106）。ハーフトーン処理とは次のような処理である。RGB画像データは、色変換を施されることによって、C、M、Y、K各色の階調データに変換される。これら各色の階調データは、階調値0から255の256階調を有するデータである。これに対し、本実施例のカラープリンタ200は、「ドットを形成する」、「ドットを形成しない」のいずれかの状態しか採り得ない。そこで、256階調を有する各色の階調データを、カラープリンタ200が表現可能な2階調で表現された画像データに変換する必要がある。このような階調数の変換を行う処理がハーフトーン処理である。後述するように、本実施例のプリンタドライバ12は、圧縮された色変換済みの画像データをメモリすなわちRAM106から読み出して、隣接する画素を所定の複数個ずつブロックにまとめた後、ブロック単位でハーフトーン処理を行う。こうしてブロック単位で処理することで、ハーフトーン処理を迅速に行うことが可能である。ブロック単位で行うハーフトーン処理の詳細については後述する。

#### 【0063】

こうしてハーフトーン処理を終了したら、プリンタドライバはインターレース処理を開始する（ステップS108）。インターレース処理とは、ドットの形成有無を表す形式に変換された画像データを、ドットの形成順序を考慮しながらカラープリンタ200に転送すべき順序に並べ替える処理である。プリンタドライバ12は、インターレース処理を行って最終的に得られた画像データを、印刷データとしてカラープリンタ200に出力する（ステップS110）。カラープリンタ200は、印刷データに従って、各色のインクドットを印刷媒体上に形成する。その結果、画像データに対応したカラー画像が印刷媒体上に印刷される。

#### 【0064】

以上に説明した画像処理では、色変換済みの画像データをメモリに一旦記憶しておき、ハーフトーン処理に際しては、必要な画像データをメモリから読み出して処理している。このため、迅速化のためにブロック単位で処理を行っているにもかかわらず、ハーフトーン処理が複雑化することがない。画像データは圧縮された状態でメモリに読み書きされるので、色変換済みの画像データを書き込む動作も、ハーフトーン処理のためにメモリから読み出す動作も、迅速に実行することができる。もっとも、こうして画像データを圧縮した状態で読み書きする場合、データの圧縮および解凍する処理が必要となるが、以下に説明するように、隣接する画素を所定の複数個ずつブロックにまとめて、ブロック内でのエッジの有無に応じて画像データを圧縮しているので、画像データの圧縮も解凍も極めて迅速に行うことが可能である。このため、本実施例の画像処理では、処理内容を複雑化させることなく、迅速な処理が可能となっている。

#### 【0065】

B-3. 第1実施例の色変換・圧縮処理：

図5は、第1実施例の色変換・圧縮処理の流れを示すフローチャートである。かかる処理は、コンピュータ100が内蔵したCPU102の機能を用いて実行する処理である。以下、フローチャートに従って説明する。

#### 【0066】

第1実施例の色変換・圧縮処理を開始すると、先ず初めに、隣接する画素を所定の複数個ずつまとめて、第1のブロックを生成する（ステップS200）。図

6は、複数の画素をまとめて第1のブロックを形成している様子を概念的に例示した説明図である。図6(a)は、縦横2列ずつの4つの画素Pa、Pb、Pc、Pdをまとめることにより、第1のブロックを生成している様子を示している。また図6(b)は、横に並んだ2つの画素Paおよび画素Pbを、第1のブロックとしてまとめている様子を示している。もちろん、第1のブロックを生成する方法は、こうした例示に限られるものではなく、互いに所定の位置関係にある複数の画素をまとめて第1のブロックとすることができる。尚、説明の便宜から、以下では、第1のブロックは、図6(a)に示すように4つの画素で構成されている場合について説明する。

#### 【0067】

第1のブロックを生成したら、該ブロック内の各画素について、画像データを色変換する処理を行う(ステップS202)。今、第1のブロックは図6(a)に示すように4つの画素で構成されているから、これら4つの画素のRGB画像データのそれぞれを、C、M、Y、K各色の階調データに変換するのである。こうした変換は、色変換テーブル(LUT)と呼ばれる3次元の数表を参照することによって迅速に行うことができる。

#### 【0068】

図7は、LUTを概念的に示した説明図である。画像データが1バイトデータであるとする、RGB各色の階調値は0から255の値を取りうるので、RGB画像データは、図7に示すように、直交3軸をそれぞれR軸、G軸、B軸とする一辺の長さ255の立方体の内部の座標として表すことができる。このような立方体を色立体と呼ぶ。LUTは色立体を格子状に細分し、各格子点にCMYKの各色の階調値を記憶した数表である。図5のステップS202においては、このようなLUTを参照することにより、第1のブロックを構成する各画素のRGB画像データを、迅速に色変換する。

#### 【0069】

次いで、色変換することによって発生したC、M、Y、K各色の中から、圧縮しようとする色を1つ選択する(ステップS204)。ついで、選択した色の階調値について、第1のブロック内の所定画素間での階調差を算出する処理を行う

(ステップ S206)。こうして階調差を求めておけば、ブロック内でのエッジの有無を容易に判断することができる。これについて、図8を参照しながら説明する。

#### 【0070】

図8(a)は、ブロック内の所定画素間の階調差を算出する様子を概念的に示した説明図である。例えば、図5のステップS204において、圧縮しようとする色としてC色を選択したものとすると、図8に示すように、ブロック内の左上の画素Paと右上の画素Pbとの間で、C色についての階調差を算出する。同様に、左上の画素Paと左下の画素Pcとの間の階調差、および画素Paと右下の画素Pdとの間でのC色についての階調差を、それぞれ算出する。画素Paと画素Pbとの階調差は主走査方向(図8上では左右方向)への階調値の変化量を示している。同様に、画素Paと画素Pcとの間の階調差は副走査方向(図8上では上下方向)への階調値の変化量を示し、画素Paと画素Pdとの間の階調差は斜め方向への変化量を示している。エッジの部分では階調差が大きくなっていると考えることができるから、こうして求めた階調差が、縦横斜めのいずれの方向についても所定の閾値以下であれば、C色についてはブロック内にエッジが存在しないと判断することができる。これに対して、いずれか一つの方向でも、階調差が所定の閾値より大きい場合は、C色についてはブロック内にエッジが存在していると判断することができる。

#### 【0071】

エッジを検出する様子を、図8(b)に示した具体例を用いて説明する。図8(b)中で、2行4列に並んで示された8つの小さな矩形は、それぞれが画素を示しており、矩形の中に表示された数値は、圧縮しようとして選択した色(ここではC色)についての各画素の階調値を示している。まず、図8(b)のブロックAに着目すると、主走査方向および副走査方向の階調差はいずれも階調値「1」であり、斜め方向への階調差は「0」である。いずれの方向についても階調差は小さく、従ってこのような場合は、ブロックAにはエッジが含まれていないと判断することができる。次にブロックBに着目すると、主走査方向の階調差は階調値「2」であるが、副走査方向の階調差は階調値「18」、斜め方向の階調差は

階調値「20」と比較的大きな値となる。従って、閾値の値を例えば階調値「15」程度に設定しておけば、ブロックB内のエッジを検出することができる。このように、ブロック内の所定画素間の階調差を検出することで、エッジの有無を容易に検出することができる。そこで、図5のステップS204においては、図8(a)に示すように、ブロック内の縦横斜め方向へのそれぞれの階調差を算出するのである。

#### 【0072】

こうして算出した階調差に基づいて、ブロック内でのエッジの有無を判断する(ステップS208)。すなわち、算出しておいた階調差と所定の閾値とを比較して、一つでも閾値より大きな階調差があれば該ブロック内にエッジが存在すると判断する(ステップS208: yes)。逆に、いずれの階調差も閾値より小さい場合は、そのブロック内にはエッジは存在しないと判断する(ステップS208: no)。

#### 【0073】

次いで、エッジの有無に応じてブロック内の画像データを圧縮し(ステップS210)、得られた画像データをメモリに書き込む処理を行う(ステップS212)。図9は、この様子を概念的に示す説明図である。図9(a)はブロック内のエッジが存在していない場合の処理を示し、図9(b)はブロック内にエッジが存在している場合の処理を示している。図中で、縦横2列に並んだ4つの小さな矩形のそれぞれは、ブロックに含まれている画素を示している。また、各矩形の中に表示されたData\_a、Data\_b、Data\_c、Data\_dは、圧縮しようとする色(ここではC色)についての各画素の階調値を表している。

#### 【0074】

ブロック内にエッジが存在しない場合は、各画素間の階調差が小さく、従ってC色については、いずれの画素の階調値もほぼ同じような値となっている。そこで、このような場合は、ブロック内での階調値の平均値Data\_cmを算出して、4つの画素の階調値Data\_a、Data\_b、Data\_c、Data\_dを、階調値Data\_cmに圧縮する。そして、メモリには、階調値Data\_cmと、階調値Data\_cmが4画素分の階調値を圧縮したことを表す値「1」とを書き込んでやる。このように、メモ



りに記憶された階調値が、何画素分の階調値に相当するかを表している値を、本明細書では圧縮情報と呼ぶ。図9（a）は、ブロック内に含まれる4つの階調値を、1つの階調値Data<sub>cm</sub>に圧縮して、圧縮情報「1」とともにメモリに書き込んでいる様子を概念的に表している。

#### 【0075】

これに対して、ブロック内にエッジが存在する場合は、各画素の階調値Data<sub>a</sub>、Data<sub>b</sub>、Data<sub>c</sub>、Data<sub>d</sub>を圧縮することなく、圧縮情報「0」とともにそのまま順番にメモリに書き込んでやる。圧縮情報「0」は、ブロック内の階調値が圧縮されておらず、従って、4つの階調値Data<sub>a</sub>、Data<sub>b</sub>、Data<sub>c</sub>、Data<sub>d</sub>が、ブロック内の4つの画素Pa、Pb、Pc、Pdの階調値に対応することを示している。図9（b）は、ブロック内に含まれる4つの階調値を、圧縮情報とともにそのままメモリに書き込んでいる様子を概念的に表している。

#### 【0076】

図5に示したステップS212では、ブロック内に含まれる各画素の階調値を、エッジの有無に応じて圧縮した後、圧縮情報とともにメモリに書き込む処理を行う。こうすれば、エッジを含むブロックについては圧縮情報を付加する分だけデータ量が増加してしまうが、エッジを含まないブロックについては、4画素分の階調値を1つの階調値で表現することができるので、全体として画像データを圧縮することができる。また、圧縮情報を記憶するために要する記憶容量は、各画素の階調値に比べれば僅かなものである。例えば、図9に示した例では、圧縮情報は「0」または「1」のいずれかの値しか採らず、従って1ブロックあたり1bitの記憶容量で足りる。このように圧縮情報を付加することによるデータ量の増加はわずかなものに過ぎず、全体としては画像データを大きく圧縮することが可能である。

#### 【0077】

尚、圧縮情報と階調値とをメモリに書き込むにあたっては、本実施例では、メモリ上に圧縮情報を記憶するための領域と階調値を記憶するための領域とを確保しておき、それぞれの領域に圧縮情報および階調値を書き込んでいる。もちろん、このような方式に限られるものではなく、例えば、圧縮情報の後に続けて階調

値を書き込むなどして、メモリ上の同じ領域に書き込むこととしても良い。

#### 【0078】

以上のようにして、圧縮しようとして選択した色（ここではC色）についての各画素の階調値と階調値とをメモリに書き込んだら、未選択の色が残っているか否かを判断する（ステップS214）。ここでは、まだC色についてしか選択していないので、ステップS214でyesと判断する。そして、ステップS204に戻って新たな色を選択し、上述した一連の処理を行う。以上のような処理を繰り返し、すべての色を選択し終わったら（ステップS214：no）、1つのブロックについて色変換・圧縮処理を行ったことになる。

#### 【0079】

こうして1つのブロックについての処理が終わったら、未処理の画素が残っているか否かを判断する（ステップS216）。そして、未処理の画素が残っていれば（ステップS216：no）、ステップS200に戻って新たなブロックを生成し、このブロック内の各画素について、続く一連の処理を行う。こうしてすべての画素についての処理を終了したら（ステップS216：yes）、図5に示す色変換・圧縮処理を終了して、図4の画像処理に復帰する。画像処理では、色変換・圧縮処理（図4のステップS104）に続いて、ハーフトーン処理（図4のステップS106）を行う。本実施例のハーフトーン処理では、圧縮された画像データをメモリから読み出した後、ブロック単位でハーフトーン処理を行うことで、迅速な処理が可能となっている。以下、本実施例のハーフトーン処理について説明する。

#### 【0080】

##### B-4. ハーフトーン処理の概要：

図10は、本実施例の画像処理において、ブロック単位で行うハーフトーン処理の流れを示すフローチャートである。かかる処理も、コンピュータ100に内蔵されたCPU102の機能を用いて実行される。尚、ハーフトーン処理は、色変換によって得られたC、M、Y、Kの各色ごとに行われるが、説明が煩雑となることを避けるために、以下では、色を特定せずに説明することで、各色について同様の処理を行っていることを表すものとする。

## 【0081】

処理を開始すると、まず初めに、ハーフトーン処理を行うブロックを設定する(ステップS300)。このブロックは、所定の位置関係で互いに隣接した複数の画素から構成されている。尚、図5を用いて前述したように、本実施例の画像処理では、色変換・圧縮処理においてもブロック単位で処理を行っている。この色変換・圧縮処理で処理するブロックと、以下に説明するハーフトーン処理で処理するブロックとは、必ずしも一致している必要はない。そこで、本明細書では、色変換・圧縮処理において設定するブロックを第1のブロックと呼び、ハーフトーン処理において設定するブロックを第2のブロックと呼んで区別することにすることがあるものとする。

## 【0082】

ハーフトーン処理においても、第2のブロックは種々の形態を取ることができる。図11は、第2のブロックが取り得る種々の形態を例示した説明図である。図中に示した小さな矩形は、それぞれが画素を表しており、斜線が付された複数の画素が、第2のブロックとしてまとめられた画素である。図中では、第2のブロックを破線で囲って表している。説明の便宜から、以下では、図11(a)に示すように、縦横2列ずつの4つの画素を第2のブロックとして設定するものとして説明する。

## 【0083】

こうして、ハーフトーン処理しようとする第2のブロックを設定したら、前述した色変換・圧縮処理において記憶しておいた圧縮情報と画像データとをメモリから読み出して、第2のブロックとして解凍する処理を行う(ステップS302)。例えば、圧縮情報の値が「1」である場合は、色変換後の画像データが圧縮されてメモリに記憶されていることを示している(図9(a)を参照のこと)。そこで、図11(a)に示した第2のブロック内の4つの画素Pa、画素Pb、画素Pc、画素Pdに、メモリから読み出した階調値Data<sub>cm</sub>を設定してやる。こうすれば、圧縮して記憶された画像データから、第2のブロックの画像データを速やかに解凍することができる。圧縮されて記憶されているブロックは、もともと画素間の階調差が小さいブロックであるから、こうしてすべての画素に同一

の階調値Data\_cmを設定しても、画質が大きく悪化することはない。

#### 【0084】

逆に、メモリから読み出した圧縮情報の値が「0」である場合は、画像データは圧縮されることなく、そのままメモリに記憶されていることを示している（図9（b）を参照のこと）。そこで、メモリから4画素分の画像データを読み出して、これら画像データを、第2のブロック内の4つの画素Pa、画素Pb、画素Pc、画素Pdの階調値として、それぞれ設定してやる。

#### 【0085】

以上に説明したように、ステップS302では、圧縮情報を用いることによりきわめて簡単な方法で画像データを解凍することができる。このため、メモリから圧縮された画像データを読み出して、第2のブロックの画像データを迅速に解凍することが可能である。

#### 【0086】

尚、ここでは、第1のブロックと第2のブロックとが一致しているものとしているが、前述したように、これらブロックは一致している必要はない。一致していない場合は、第2のブロックに含まれるすべての画素について画像データを設定し終わるまで、メモリに記憶された圧縮情報と画像データとを読み出してやればよい。従って、第2のブロック中の一部の画素は圧縮された状態でメモリに記憶されており、他の画素は圧縮されずに記憶されているといった場合も起こり得る。

#### 【0087】

ついで、こうして解凍した第2のブロック内の各画素について、ブロック単位でドット形成の有無を判断する処理を行う（ステップS304）。この処理の詳細については、別図を用いて後述する。

#### 【0088】

第2のブロック内の各画素について、ドット形成の有無を判断したら、該判断によってブロック内で生じた階調表現の誤差を、周辺の画素に拡散する処理を行う（ステップS306）。ブロック内で生じた階調表現の誤差を求める方法については後述する。図12は、ブロック内で生じた誤差を周辺の画素に拡散してい

る様子を概念的に示す説明図である。図中に示した小さな矩形は画素を表しており、破線で示した大きな矩形は第2のブロックを表している。斜線が付されたブロックが、ドット形成の有無が判断されたブロックである。図10のステップS308では、このブロック全体で発生した階調表現の誤差を、周辺の画素に向かって所定の比率で分配する処理を行う。一例として、ブロック内で生じた誤差を周辺の各画素に分配する比率を、図13に例示した。図13(a)は、判断を行ったブロックに隣接している6つの画素に分配する場合を示し、図13(b)は、判断を行ったブロックに隣接するブロック内の各画素に分配する場合を示している。図12に示した黒い矢印は、図13(a)に例示された比率に従って、ブロックで生じた誤差を周辺の画素に拡散している様子を概念的に表している。こうして拡散された誤差は各画素毎に蓄積され、各画素に対応付けられた状態でコンピュータ100のRAM106に記憶される。

#### 【0089】

以上のようにして、ブロック内で生じた誤差を拡散したら、全画素についての処理を終了したか否かを判断する(ステップS308)。そして、未処理の画素が残っていれば(ステップS308: no)、ステップS300に戻ってブロックの位置を移動させ、設定した第2のブロックについて、上述した一連の処理を行う。こうして全ての画素についての処理を終了したら(ステップS308: yes)、図10に示したハーフトーン処理を終了して図4の画像処理に復帰する。

#### 【0090】

以下では、ブロック単位で各画素についてのドット形成の有無を判断する処理(ブロック単位2値化処理)について説明する。図14は、本実施例のハーフトーン処理中で、ブロック単位でドット形成の有無を判断する処理の流れを示したフローチャートである。

#### 【0091】

図14に示すブロック単位2値化処理を開始すると、先ず初めに、処理しようとするブロックの中から、ドット形成の有無を判断する画素(着目画素)を1つ設定する(ステップS400)。つまり、前述したように、複数の画素をまとめ

て第2のブロックとしているから、これらの画素を所定の順序で1画素ずつドット形成の有無を判断することとして、着目画素を1つ選択するのである。図11(a)に示したように、第2のブロックは隣接する4つの画素から構成されているとしている。ここでは、これら4つの画素を、ブロック内の左上にある画素から右下の画素に向かって画素Pa、画素Pb、画素Pc、画素Pdの順番にドット形成有無を判断することとする。そこで、先ず初めに、ステップS400では、左上隅にある画素Paを着目画素として設定する。

#### 【0092】

次いで、設定した着目画素（ここでは、画素Pa）についての補正データCxを算出する（ステップS302）。補正データCxは、着目画素の画像データと着目画素に拡散されて記憶されている誤差（拡散誤差）とを加算することによって求めることができる。処理している第2のブロックの各画素についての画像データは、図10に示したハーフトーン処理中のステップS302において、圧縮されてメモリに記憶されているデータを読み出して既に解凍され、CPU102に内蔵されたレジスタあるいはキャッシュに記憶されている。また、着目画素についての拡散誤差は、図10のステップS306中で、画素毎に蓄積されて記憶されている。図14に示したステップS402においては、着目画素についての画像データと拡散誤差とを加算することによって、補正データCxを算出する処理を行う。

#### 【0093】

着目画素Paの補正データCxを算出したら、得られた補正データCxと所定の閾値thとの大小関係を判断する（ステップS404）。そして、補正データCxの方が大きければ着目画素（ここでは画素Pa）にドットを形成すると判断し（ステップS406）、そうでなければ、着目画素にはドットを形成しないと判断する（ステップS408）。判断の結果は、各画素についての判断結果を示す変数としてRAM106に蓄えられる。

#### 【0094】

こうして着目画素についてのドット形成有無を判断したら、判断に伴って着目画素に生じた階調誤差を算出する（ステップS410）。階調誤差は、ドットを

形成したこと、あるいはドットを形成しなかったことによって着目画素に表現される階調値（以下では、この階調値を結果値と呼ぶ）を、着目画素の補正データ  $C_x$  から減算することによって算出することができる。

#### 【0095】

着目画素で生じた階調誤差を算出したら、処理中のブロック内の全画素の処理を終了したか否かを判断する（ステップ S 4 1 2）。そして、未処理の画素が残っている場合は（ステップ S 4 1 2 : no）、これら未処理の画素中で着目画素周辺にある画素に誤差を拡散させる処理を行う（ステップ S 4 1 4）。この処理について、図 1 5 を参照しながら説明する。

#### 【0096】

図 1 5 は、処理中の第 2 のブロック内の画素から着目画素を 1 つずつ選択してドット形成の有無を判断することにより、ドット形成の有無をブロック単位で判断している様子を模式的に示した説明図である。図中の 4 つの正方形は、処理中の第 2 のブロックを構成する画素  $P_a$ 、画素  $P_b$ 、画素  $P_c$ 、画素  $P_d$  をそれぞれ示している。また、画素  $P_a$  を示す正方形の中に示した  $DT_a$  は、メモリから読み出されて解凍された画像データを示しており、また、画素  $P_a$  中に示した  $ED_a$  は、画素  $P_a$  に拡散された蓄積されている拡散誤差を示している。同様に、画素  $P_b$  中に示した  $DT_b$  および  $ED_b$  はそれぞれ画素  $P_b$  の画像データおよび拡散誤差を示し、画素  $P_c$  中の  $DT_c$  および  $ED_c$  はそれぞれ画素  $P_c$  の画像データおよび拡散誤差  $ED_c$  を、画素  $P_d$  中の  $DT_d$  および  $ED_d$  は画素  $P_d$  の画像データおよび拡散誤差を示している。前述したように、これら各画素の画像データは、RAM 106 から読み出されて解凍された後、CPU 102 に内蔵されたレジスタあるいはキャッシュに記憶されている。また、そのブロックについての処理を開始した直後では、各画素の拡散誤差は、RAM 106 に記憶されている。

#### 【0097】

前述したように、ブロック単位 2 値化処理を開始すると、先ず初めにブロック内で左上の画素  $P_a$  が着目画素として設定される（図 1 4 のステップ S 4 0 0 参照）。そして、画素  $P_a$  についての補正データ  $C_x$  を算出し、閾値  $t_h$  と比較す

ることによってドット形成の有無を判断する（図14のステップS402, S404参照）。画素Paについてドット形成の有無を判断した時点では、図15（a）に示すように、同じブロック内には画素Pbと画素Pcと画素Pdの3つの画素が未判断画素として残っている。そこで、画素Paで生じた階調誤差を、これら3つの画素に1/3ずつ均等に分配して、これら誤差を、各画素毎に記憶されている拡散誤差に加算する処理を行う。例えば図15（a）に示すように、画素Pbに既に記憶されている拡散誤差をEDbとすると、画素Paで生じた階調誤差は、その1/3の階調値が画素Pbに分配されて、拡散誤差EDbに加算される。その結果、画素Pbに記憶されていた拡散誤差EDbは、画素Paからの誤差が加算された新たな拡散誤差EDbに更新される。更新された拡散誤差は、CPU102のレジスタあるいはキャッシュに記憶される。

#### 【0098】

他の画素Pcおよび画素Pdについても同様な処理を行う。簡単に説明すると、画素Paからの誤差が画素Pcに拡散されると、画素Pcに記憶されていた拡散誤差EDcと画素Paからの誤差とが加算され、新たな拡散誤差EDcとして画素Pcに記憶される。また、画素Pdについても同様に、画素Pdに記憶されていた拡散誤差EDdと画素Paからの誤差とが加算され、新たな拡散誤差EDdとして画素Pdに記憶される。図15（a）中で画素Paから他の3つ画素に向かって表示されている白抜きの矢印は、画素Paで生じた誤差がこれら3つの画素に拡散されることを模式的に示したものである。こうして画素Paで生じた階調誤差が拡散されて更新された各画素の拡散誤差は、CPU102のレジスタあるいはキャッシュに記憶される。尚、階調誤差は、必ずしも周辺の未判断画素に均等に分配する必要はなく、各画素に所定の割合で分配することも可能である。

#### 【0099】

以上のようにして画素Paで生じた階調誤差をブロック内の周辺画素に拡散したら、図14のステップS414の処理を終了してステップS400に戻り、新たな着目画素を設定する処理を行う。画素Paについては既にドット形成の有無を判断したから、ステップS400では、今度は画素Paの右隣の画素Pbを着



目画素として設定する。こうして設定した新たな着目画素について、上述した処理とほぼ同様の処理を行う。以下、画素Paにおける処理との相違点を中心に簡単に説明する。

#### 【0100】

画素Pbについての処理を行う場合にも画素Paにおける場合と同様に、先ず初めに、補正データCxを算出する(ステップS402)。ここで、画素Pbに記憶されている拡散誤差EDbには、先に画素Paで生じた階調誤差が拡散されて、新たな拡散誤差EDbとして更新されている。そこで、画素Pbについての画像データDTbと、画素Paからの階調誤差が拡散された新たな拡散誤差EDbとを加算することにより、画素Pbについての補正データCxを算出する。ここで、画像データDTbおよび拡散誤差EDbは、いずれもCPU102内のレジスタあるいはキャッシュに記憶されているので、極めて高速に読み出して補正データCxを算出することができる。こうして算出した補正データCxと所定の閾値thとの大小関係を判断し(ステップS404)、補正データCxの方が大きければ画素Pbにはドットを形成すると判断し(ステップS406)、そうでなければ画素Pbにはドットを形成しないと判断する(ステップS408)。次いで、画素Pbについて発生した階調誤差を算出する(ステップS410)。階調誤差は、画素Paの場合と同様に、画素Pbの補正データCxから結果値を減算することによって算出することができる。

#### 【0101】

着目画素Pbで生じた階調誤差を算出したら、処理中の第2のブロック内のすべての画素について処理を終了したか否かを判断し(ステップS412)、未処理の画素が残っていれば、算出しておいた階調誤差を、これら未処理の画素に拡散させる処理を行う(ステップS414)。前述したように、ブロック内の画素Paについては既に判断を終了しているから、未判断画素として残っているのは画素Pcおよび画素Pdの2つの画素である。そこで、画素Pbで生じた階調誤差はこれら2つの画素に拡散させることになる。再び、図15を参照しながら、画素Pbからの誤差を拡散させる処理について説明する。

#### 【0102】

図15 (b) は、画素Pb で生じた階調誤差を拡散している様子を概念的に示した説明図である。図中で、画素Pa に斜線が施されているのは、画素Pa については既にドット形成の有無を判断済みであることを模式的に示したものである。画素Pb についてドット形成の有無を判断した時点では、処理ブロック内には未判断画素として画素Pc、画素Pd の2つの画素が残っている。そこで、画素Pb で生じた階調誤差は、図15 (b) に示したように、これらの画素Pc および画素Pd にそれぞれ、 $1/2$ ずつ拡散させる。図中で、画素Pb から画素Pc および画素Pd に向かって伸びる白抜きの矢印は、画素Pb で生じた階調誤差がこれら画素に拡散されていることを模式的に示している。こうして誤差が拡散される結果、各画素の拡散誤差は、拡散されてきた誤差が加算された新たな拡散誤差に更新されて、CPU102のレジスタあるいはキャッシュに記憶される。尚、ここでも、階調誤差は処理ブロック内の未判断画素に均等に拡散されるものとしているが、もちろん、均等に拡散させるのではなく、所定の割合で誤差を拡散させることとしても構わない。

#### 【0103】

こうして画素Pb で生じた階調誤差を周辺の画素に拡散したら、再び図14のステップS400に戻って、今度は着目画素を画素Pc に移動させた後、着目画素Pc について同様の処理を行う。以下では、かかる処理について、図15 (c) を参照しながら説明する。画素Pc についての処理を開始すると、先ず初めに画素Pc の画像データDTc と拡散誤差EDc とを加算して補正データCx を算出する。ここで、画素Pc についてのドット形成有無を開始した時点では、画素Pc の拡散誤差EDc は、元々記憶されていた拡散誤差に加えて、画素Pa からの誤差と画素Pb からの誤差とが加算され、新たな拡散誤差EDc として更新されてCPU102のレジスタあるいはキャッシュに記憶されている。このことから、画素Pc の補正データCx の算出に際しては、画素Pc の画像データDTc に、新たな拡散誤差EDc が算出されることになる。こうして得られた補正データCx と、所定の閾値th とを比較することによって、画素Pc についてのドット形成有無を判断する。こうして画素Pc についてのドット形成有無を判断した時点では、処理中のブロック内に残る未判断画素は、図15 (c) に示すように

画素Pdのみである。そこで、画素Pcで生じた階調誤差は全て画素Pdに拡散され、画素Pdに元々記憶されていた拡散誤差に加算されて、新たな拡散誤差EDdとして更新される。

#### 【0104】

着目画素Pcで生じた階調誤差を拡散したら、図14のステップS400に戻って着目画素を画素Pdに移動した後、画素Pdについてのドット形成有無の判断を開始する。すなわち、画素Pdに記憶されている拡散誤差EDdと画素Pdの画像データDTdとを加算して補正データCxを算出する(ステップS402)。次いで、補正データCxと閾値thとを比較し(ステップS404)、補正データCxの方が大きい場合は画素Pdにはドットを形成する(ステップS406)と判断し、そうでない場合は画素Pdにはドットを形成しないと判断する(ステップS408)。

#### 【0105】

こうして画素Pdについてのドット形成の有無を判断したら、画素Pdで生じた階調誤差を算出した後(ステップS410)、処理中のブロック内の全画素の処理を終了したか否かを判断する(ステップS412)。図15(d)に示したように、画素Pdについて処理を開始した時点で、もはや処理中のブロック内には未判断の画素は残っておらず、画素Pdについての判断が終了した後のステップS412においては、ブロック内の全画素についての処理が終了したと判断される。そこで、図14に示したブロック単位2値化処理を終了して、図10を用いて前述したブロック単位で行うハーフトーン処理に復帰する。前述したように、図10に示したハーフトーン処理では、ブロック単位2値化処理から復帰すると、処理した第2のブロックで生じた階調誤差を周辺のブロックに拡散する処理を行う(図10のステップS306)。

#### 【0106】

ここで、処理した第2のブロックで生じた階調表現の誤差としては、ブロック単位2値化処理のステップS410において画素Pdについて求めた階調誤差の値を用いることができる。これは、着目画素で生じた階調誤差をブロック内の未判断画素に拡散しながらドット形成有無の判断を行っているために、ブロック内

で生じた階調誤差が、最後に判断する画素Pdに拡散誤差EDdとして蓄積されることによる。以下、図15を参照して詳しく説明する。

#### 【0107】

画素Paで生じた階調誤差は、処理中のブロック内の画素Pb、画素Pc、画素Pdの3つの画素に拡散される。画素Pbのドット形成有無の判断に際しては、画素Paからの誤差を考慮して、画素Pbに拡散されている誤差ができるだけ解消されるようにドット形成の有無が判断される。仮に、画素Pbについてドット形成有無を判断することにより、画素Pbに拡散されている誤差を解消することができたものとする。この場合は、画素Pbでは新たな階調誤差は発生せず、また、画素Pbから画素Pc、画素Pdに拡散される誤差も生じない。他の2つの画素Pc、画素Pdについても同様に、ドット形成の有無を判断したことで、それぞれの画素に拡散されている誤差を解消することができたものとする、これらの画素では階調誤差は生じない。結局、この場合は、画素Paで生じた階調誤差を、周辺の未判断画素Pb、画素Pc、画素Pdで解消したこととなり、処理ブロック全体としては階調誤差の発生を避けることができる。

#### 【0108】

ここでは、画素Pbのドット形成有無の判断によって、画素Pbでの誤差を解消するものとしたが、画素Pbで新たな階調誤差が発生した場合には、この誤差は周辺の未判断画素Pcおよび画素Pdに拡散されて、その結果、これらの画素には、画素Paで生じた階調誤差と画素Pbで生じた階調誤差とが拡散されることになる。しかし、これら画素Pc、画素Pdについてドット形成の有無を判断することで、それぞれの画素に拡散された誤差を解消することができれば、依然としてブロック全体としてみれば階調誤差は生じない。

#### 【0109】

画素Pcについてドット形成有無を判断することで、画素Pcに拡散されている誤差を解消することができない場合も、全く同様にして、画素Pdで誤差を解消することができれば、ブロック全体としては階調誤差が生じることはない。換言すれば、画素Pdで解消しきれずに残った誤差が、ブロック全体で生じた階調誤差となる。このように、着目画素で生じた階調誤差をブロック内の未判断

画素に拡散しながらドット形成有無を判断していくと、各画素で解消しきれずに残った階調誤差が後続する画素に集約されていくので、ブロック内で最後に判断する画素Pdで生じた階調誤差を、処理中のブロック全体で生じた階調誤差として用いることができるのである。

#### 【0110】

尚、以上の説明では、処理中のブロック内で最後にドット形成有無を判断する画素Pdで生じた階調誤差の値を、ブロック全体で生じた階調誤差として用いるものとしたが、もちろん、ブロックを構成する各画素毎に階調誤差を算出し、これら階調誤差の総和をブロック全体としての階調誤差としても構わない。

#### 【0111】

また、図15に示した例では、第2のブロックを構成する各画素のドット形成有無を判断するに際しては、画素Pa、画素Pb、画素Pc、画素Pdの順番で判断を行ったが、必ずしもこの順番で判断する必要はなく、例えば、画素Pa、画素Pc、画素Pb、画素Pdの順番でドット形成有無を判断しても構わない。第2のブロック内で着目画素を設定する順番は、良好な画質が得られる順番を選択しておくことができる。

#### 【0112】

以上に説明したように本実施例の画像処理では、色変換後の画像データに基づいて、第2のブロック内にエッジが存在するか否かを検出する。そして、エッジの有無によって画像データを圧縮し、圧縮後の画像データを一旦メモリに記憶する。続くハーフトーン処理では、メモリから必要な画像データを読み出して、ブロック単位でハーフトーン処理を行う。ハーフトーン処理をブロック単位で行えば、ブロック内の各画素については、誤差を拡散する処理あるいは補正データCxを算出してドット形成の有無を判断する処理を迅速に行うことができるので、ハーフトーン処理を高速化することができる。

#### 【0113】

また、ハーフトーン処理をブロック単位で行った場合、処理が複雑化し易いが、本実施例では、メモリに画像データを圧縮して記憶しておき、必要に応じて読み出してハーフトーン処理を行っている。このように必要なデータを読み出しな

がらハーフトーン処理を行えば、処理が複雑化することを効果的に回避することができる。

#### 【0114】

加えて、ハーフトーン処理を行う画像データ（色変換後の画像データ）は圧縮されているので、メモリに書き込む動作もメモリから読み出す動作も極めて迅速に行うことができる。更に本実施例の画像処理では、ブロック内でのエッジの有無に応じて画像データを圧縮しているので、画像データの圧縮および解凍も極めて迅速に且つ効率良く行うことができる。すなわち、ブロック内でエッジが検出されなかった場合は、ブロック内の各画素の階調値を平均階調値によって代表させることによって画像データを圧縮し、解凍に際しては各画素に平均階調値を設定する。また、ブロック内にエッジが検出された場合は画像データの圧縮は行わない。このようにエッジが検出された場合は圧縮を行わないとしても、画像データ全体としては、実用上十分な圧縮率を確保することができる。これは、ブロックを極端に大きなものに設定しない限り、エッジが検出されるブロックはさほど多くはないことと、エッジが検出されなかったブロックについての圧縮率が高いことによる。こうしてエッジの有無に応じて圧縮および解凍しているために、圧縮も解凍も極めて迅速に行うことができる。

#### 【0115】

以上に説明したように、本実施例の画像処理によれば、処理を複雑化させることなく、画像処理全体としての処理速度を向上させることが可能となる。

#### 【0116】

##### C. 第2実施例：

上述した第1実施例では、エッジの検出は色変換後の画像データに基づいて行った。しかし、色変換前の画像データに基づいてエッジを検出することで、エッジを検出する処理をより簡素なものとすることもできる。以下では、こうした第2実施例について説明する。

#### 【0117】

図16は、第2実施例の色変換・圧縮処理の流れを示すフローチャートである。以下ではフローチャートを参照しながら、第1実施例との相違点を中心に説明

する。

#### 【0118】

第2実施例においても、色変換・圧縮処理を開始すると、先ず初めに、隣接する画素を所定の複数個ずつまとめて、第1のブロックを生成する（ステップS500）。ここでは、前述した第1実施例と同様に、第1のブロックは、図6（a）に示すように、縦横2列ずつに並んだ4つの画素で構成されているものとして説明する。

#### 【0119】

第1のブロックを生成したら、該ブロック内の所定画素間で、各画素についてRGB画像データの階調差を算出する処理を行う（ステップS502）。すなわち図8（a）に示したように、画素Paと画素Pbとの間、画素Paと画素Pcとの間、画素Paと画素Pdとの間のそれぞれについて、RGB各色毎に階調差を算出するのである。前述した第1実施例の色変換・圧縮処理では、このように、RGB画像データを色変換して得られたC、M、Y、Kの各色についての階調差を算出したが、第2実施例では、色変換前のRGB各色について階調差を算出する点が大きく異なっている。

#### 【0120】

こうして算出した階調差に基づいて、ブロック内でのエッジの有無を判断する処理を行う（ステップS504）。ステップS502において所定画素間で算出したRGB各色の階調差の中で、所定の閾値より大きなものが1つでも含まれていれば、そのブロックにはエッジが含まれていると判断する。逆に、いずれの階調差も閾値より小さい場合には、そのブロックにはエッジは含まれていないと判断する。

#### 【0121】

そして、エッジが含まれていない場合は（ステップS504：no）、ブロック内のRGB画像データを圧縮し（ステップS506）、圧縮したRGB画像データをCMYK各色による画像データに色変換する（ステップS508）。RGB画像データの圧縮は、第1のブロックに含まれる各画素についての画像データの平均値を、各色毎に算出することによって行う。例えば、ブロック内の各画素

$P_a$ 、 $P_b$ 、 $P_c$ 、 $P_d$  の R 画像データの階調値が、それぞれ  $R_a$ 、 $R_b$ 、 $R_c$ 、 $R_d$  とすると、R 画像データは、階調値  $R_{ave}$  に圧縮される。ここで、

$$R_{ave} = (R_a + R_b + R_c + R_d) / 4$$

である。G 画像データ、B 画像データについても同様に、ブロック内の各画素  $P_a$ 、 $P_b$ 、 $P_c$ 、 $P_d$  の G 画像データの階調値を  $G_a$ 、 $G_b$ 、 $G_c$ 、 $G_d$  として、B 画像データの階調値を  $B_a$ 、 $B_b$ 、 $B_c$ 、 $B_d$  とすると、G 画像データは、階調値  $G_{ave}$  に圧縮され、B 画像データは、階調値  $B_{ave}$  に圧縮される。ここで、 $G_{ave}$  および  $B_{ave}$  は、それぞれ

$$G_{ave} = (G_a + G_b + G_c + G_d) / 4$$

$$B_{ave} = (B_a + B_b + B_c + B_d) / 4$$

として求められる。

#### 【0122】

続くステップ S508 では、こうして圧縮された RGB 画像データ ( $R_{ave}$ 、 $G_{ave}$ 、 $B_{ave}$ ) について、図 7 に示した LUT を参照しながら色変換する。第 2 実施例では、第 1 のブロック内にエッジが存在しない場合には、こうして RGB 画像データを圧縮した後に色変換しているのので、画素毎に色変換を行う必要が無く、迅速に色変換することができる。また、エッジが存在しないブロックでは、画素間での RGB 画像データがほぼ同じ階調値となっているから、各画素についての RGB 画像データを色変換して得られた CMYK 各色の階調値も、ほぼ同じ階調値になると考えられる。従って、こうしたブロックについては、圧縮した RGB 画像データを色変換することとしても、ほとんど画質が悪化することはない。

#### 【0123】

一方、ステップ S504 においてブロック内にエッジが存在すると判断された場合は (ステップ S504: yes)、該ブロックに含まれる各画素毎に、RGB 画像データを CMYK 画像データに色変換する (ステップ S510)。

#### 【0124】

尚、以上の説明では、ブロック内の所定画素間での全ての階調差を各色毎に一旦算出し (ステップ S502)、その後ステップ S504 において、算出した階



調差と閾値を比較してエッジの有無を検出するものとして説明した。しかし、これは、理解の便宜を考慮して説明の簡素化を図ったためであり、必ずしもブロック内の全ての階調差を算出する必要はない。すなわち、所定の閾値を超える階調差が1つでも見出されたら、他の階調差については算出することなく、そのブロックにはエッジ存在すると判断して、ステップS510の処理を行うこととしても良い。

#### 【0125】

以上のようにして、圧縮したRGB画像データあるいは圧縮していないRGB画像データの色変換を終了したら、得られたCMYK各色の画像データを、圧縮情報とともにメモリすなわちRAM106に記憶する（ステップS512）。圧縮情報は、第1のブロックにエッジが存在していない場合（圧縮したRGB画像データを色変換した場合）には、画像データが圧縮されていることを表す値「1」を記憶する。逆に、第1のブロックにエッジが存在する場合は、画像データが圧縮されていないことを表す値「0」を、圧縮情報として記憶する。

#### 【0126】

尚、前述した第1実施例では、C、M、Y、K各色毎にエッジの検出および画像データの圧縮を行っているので、同一ブロック内でも、色によって圧縮状態が異なっている場合が起こり得る。このため圧縮情報は、ブロック毎に、そして各色毎に設定して記憶されている。これに対して第2実施例では、エッジの検出と圧縮とをブロック単位で行っているので、同一ブロックで有れば、圧縮状態はCMYK各色いずれも同じとなっている。従って、ブロック毎に1つずつ圧縮情報を設定して記憶することも可能である。

#### 【0127】

こうして1つのブロックについての処理が終わったら、未処理の画素が残っているか否かを判断する（ステップS514）。そして、未処理の画素が残っていれば（ステップS514：no）、ステップS500に戻って新たなブロックを生成し、このブロック内の各画素について、続く一連の処理を行う。こうしてすべての画素についての処理を終了したら（ステップS514：yes）、図16に示す第2実施例の色変換・圧縮処理を終了して、図4の画像処理に復帰する。

## 【0128】

こうした第2実施例の色変換・圧縮処理においては、エッジの存在しないブロックについては、1回だけ色変換すればよい。従って、ブロック内の全画素について色変換する第1実施例の色変換・圧縮処理よりも、処理の高速化を図ることができる。また、エッジの検出に際しても、RGB各色についての階調差を算出して、閾値より大きな階調差が1つでも見出されたら、残りの階調差は算出することなく、次の処理を開始することができる。この点においても、第2実施例の色変換・圧縮処理は、第1実施例の色変換・圧縮処理よりも処理の高速化を図ることが可能である。

## 【0129】

## D. 第3実施例:

上述した第2実施例の色変換・圧縮処理では、ブロック内の画素間でRGB各色についての階調差がいずれも小さい場合にのみ、画像データを圧縮する。こうすることにより、第1実施例に対して画像処理をより迅速化することが可能であるが、その一方で、画像データ全体としてみたときの圧縮率が小さくなる傾向がある。圧縮率があまりに小さくなってしまえば、画像データをメモリすなわちRAM106に対して読み書きするために時間がかかり、却って処理速度が低下するおそれが生じる。こうした可能性を回避するためには、色変換の前後で画像データを圧縮することが効果的である。以下では、こうした第3実施例の色変換・圧縮処理について説明する。

## 【0130】

図17は、第3実施例の色変換・圧縮処理の流れを示すフローチャートである。以下ではフローチャートを参照しながら、第1実施例および第2実施例との相違点を中心に説明する。

## 【0131】

第3実施例においても、色変換・圧縮処理を開始すると、先ず初めに、隣接する画素を所定の複数個ずつまとめて、第1のブロックを生成する(ステップS600)。ここでも前述した各種実施例と同様に、第1のブロックは、図6(a)に示すように、縦横2列ずつに並んだ4つの画素で構成されているものとして説

明する。

### 【0132】

第1のブロックを生成したら、該ブロック内の所定画素間で、各画素についてのRGB画像データの階調差を算出して（ステップS602）、エッジを検出する処理を行う（ステップS604）。かかる処理は、第2実施例におけるステップS502およびS504と同様である。以下、概要のみ説明すると、先ず、ブロック内の所定画素間について、RGB各色毎に階調差を算出し、所定の閾値と比較する。そして、閾値より大きな階調差が見出されたら、そのブロックについてはエッジを含んでいると判断する。逆に、全ての階調差が閾値より小さいことが確認されたブロックについては、エッジを含んでいないと判断する。

### 【0133】

ブロック内にエッジが存在していないと判断された場合は（ステップS604：no）、前述した第2実施例と同様に、ブロック内のRGB画像データを圧縮し（ステップS606）、圧縮したRGB画像データの色変換を行う（ステップS608）。

### 【0134】

一方、ブロック内にエッジが存在していると判断された場合は（ステップS604：yes）、各画素のRGB画像データをC、M、Y、K各色の画像データに色変換する（ステップS610）。次いで、第1実施例の色変換・圧縮処理と同様に、ブロック内の所定画素間での階調差を各色毎に算出（ステップS612）、ブロック内でのエッジの有無を各色毎に判断する（ステップS614）。すなわち、所定画素間での階調差がいずれも、所定の閾値より小さければ、その色についてはブロック内にエッジが存在しないと判断する。逆に、閾値より大きな階調差が存在する場合には、その色についてはブロック内にエッジが含まれていると判断する。

### 【0135】

こうしてブロックについてのエッジの有無を検出した結果、エッジが含まれていないと判断された色については（ステップS614：no）、ブロック内の画像データを圧縮する処理を行う（ステップS616）。画像データの圧縮は、前

述した第1実施例と同様に、圧縮しようとしている色について、ブロック内の各画素の平均階調値を算出し、各画素の階調値を平均階調値で代表させることによって行う。一方、ブロック内にエッジが含まれていると判断された色については、ブロック内の画像データを圧縮する処理はスキップする。

#### 【0136】

以上のようにして、ブロック内でのエッジの有無を検出し、エッジの有無に応じて画像データを圧縮する処理を行ったら、これら画像データを圧縮情報とともにメモリすなわちRAM106に書き込む処理を行う（ステップS618）。圧縮情報は、画像データを圧縮している色については「1」を、圧縮していない色については「0」を書き込んでやる。

#### 【0137】

こうして1つのブロックについての処理が終わったら、未処理の画素が残っているか否かを判断する（ステップS620）。そして、未処理の画素が残っていれば（ステップS620：no）、ステップS600に戻って新たなブロックを生成し、このブロック内の各画素について、続く一連の処理を行う。こうしてすべての画素についての処理を終了したら（ステップS620：yes）、図17に示す第3実施例の色変換・圧縮処理を終了して、図4の画像処理に復帰する。

#### 【0138】

以上に説明した第3実施例の色変換・圧縮処理においては、色変換前の画像データに基づいてエッジの有無を検出し、エッジを含まないブロックについては、RGB画像データを圧縮した後に色変換する。こうして、圧縮後に色変換すれば、色変換を1度に行うことができるので、迅速に色変換・圧縮処理を行うことができる。

#### 【0139】

また、エッジの検出されたブロックについては、各画素毎に色変換した後、各色毎にエッジの有無を検出する。そして、エッジの検出されなかった色については画像データの圧縮を行う。こうすれば、色変換後の画像データについて各色毎にエッジの有無を検出して圧縮しなければならないのは、色変換前にエッジが検出されたブロックだけでよいので、色変換・圧縮処理を迅速に行うことができる。

。加えて、これらブロックについては各色毎にエッジの有無を検出して圧縮している。第1実施例と同様に、十分な圧縮率を確保することが可能となる。

#### 【0140】

##### E. 変形例:

以下では、上述した各種実施例の変形例について簡単に説明する。

#### 【0141】

##### (1) 第1の変形例:

上述した各種実施例では、ブロック内でのエッジの有無を検出するために、図8に示した所定画素間での階調差を算出するものとした。もっとも、エッジを検出するために算出する階調差は、図8に示した所定画素間の階調差に限定されるものではない。例えば、図18(a)に示すように、画素Paと画素Pdとの階調差および画素Pbと画素Pcとの階調差を算出することとしても良い。こうすれば、例えば図18(b)に示すように、幅が1画素の細いエッジがブロック内を斜めに横切っていない限り、2つの階調差を算出するだけでブロック内のエッジを正しく検出することができる。

#### 【0142】

##### (2) 第2の変形例:

上述した各種実施例では、ブロック内の画像データを圧縮する場合、先ず、該ブロック内の各画素の平均階調値を算出して、各画素の階調値を平均階調値に圧縮するものとして説明した。しかし、簡便には、平均階調値の代わりに、ブロック内の1つの画素の階調値に圧縮することとしても良い。再び、図18(a)を用いて説明すると、ブロック内でエッジが検出されなかった場合は、図中で斜線を付した画素Paの階調値で、ブロック内の他の画素の階調値を代表させても良い。エッジの存在しないブロックについては、各画素の階調値はほぼ近い値を取るから、平均階調値を1つの画素の階調値で代用することも可能である。階調値を平均階調値の代わりに用いる画素は、ブロック内の所定位置の画素とすることもできるし、任意の画素を選択することとしても良い。

#### 【0143】

##### (3) 第3の変形例:

上述した各種実施例においては、ブロック内にエッジが含まれているか否かだけを判断して、エッジが含まれていなければ画像データを圧縮し、エッジが含まれていれば圧縮しないものとして説明した。すなわち、画像データの圧縮状態は、圧縮されているか圧縮されていないかの、いずれかの状態しか取らないものとして説明した。しかし、エッジの有無だけでなく、エッジの方向も判断して、それに応じて圧縮状態を切り換えることとしても良い。

#### 【0144】

図19は、エッジの方向によって種々の圧縮状態を切り換えている様子を例示した説明図である。図19(a)は、ブロック内にエッジが含まれていない場合を示している。この場合は、上述した各種実施例と同様に、各画素の画像データを1つの階調値に圧縮してやる。また、前述した各種実施例では、圧縮状態は2つの状態しか取り得なかったので、圧縮情報は1bitのデータで足りたが、図19に例示した場合は4つの圧縮状態を取り得るので、圧縮情報は2bitのデータとなる。そこで、エッジが含まれていないブロックについては、圧縮情報として値「11」を記憶してやる。

#### 【0145】

また、縦方向に並んだ2つの画素間の階調差、すなわち、画素Paと画素Pcとの間の階調差、および画素Pbと画素Pdとの間の階調差はいずれも小さいが、横方向あるいは斜め方向の画素間の階調差は必ずしも小さくない場合は、ブロック内をエッジが縦方向に走っているものと考えられる。そこで、このような場合は、図19(b)に示すように、1つのブロックを2つのサブブロックに分けて考えて、サブブロック毎に画像データを圧縮してやる。また、このときの圧縮情報として「10」を記憶してやる。

#### 【0146】

更に、横に並んだ2つの画素間での階調差は小さいが、縦方向あるいは斜め方向の画素間では必ずしも階調差が小さくない場合は、エッジがブロック内を横方向に走っていると考えられる。そこで、このような場合は、1つのブロックを、図19(c)に示すような2つのサブブロックに分割して、サブブロックごとに画像データを圧縮してやる。このときは、圧縮情報として「01」を記憶する。

## 【0147】

そして、いずれの場合に該当しない場合は、図19(d)に示すように、画像データを圧縮することなく、そのままメモリに記憶してやる。また、このときの圧縮情報は、画像データが圧縮されていないことを示す「00」を記憶する。

## 【0148】

この様に、ブロック内でのエッジの有無だけでなく、エッジが走っている方向も検出し、これに応じて画像データを圧縮してやれば、画像データの圧縮率を更に向上させることができる。また、このような圧縮を行った場合でも、上述した各種実施例と同様に、画像データの圧縮および解凍する処理を迅速に行うことができる。このため、色変換後の画像データをメモリに対して読み書きする動作が更に高速化され、画像処理全体として、より迅速な処理を行うことが可能となる。

## 【0149】

(4) 第4の変形例:

図14および図15を用いて説明したハーフトーン処理では、ブロックを構成する各画素がどのような階調値を取る場合であっても、処理内容は全く同じものとして説明した。しかし、ブロック内でも、エッジを含まないとして圧縮されていた画素間では近似した階調値を取るものと考えられる。特に、図9に示したように、エッジを含まないブロックについては複数の画素の階調値を単一の階調値に圧縮する場合、解凍された各画素の階調値は同じ値となる。このように、エッジを含まないとして圧縮されていた画素については、近似した(あるいは同一の)階調値を有するものと考えられるので、これを利用して、処理の高速化を図ることも可能である。

## 【0150】

例えば、ディザ法を用いたハーフトーン処理においてディザマトリックスを画像データの階調値に応じて使い分けたり、あるいは誤差拡散法で用いられる閾値を画像データの階調値に応じて変更する場合がある。また、誤差拡散法では、画像データの階調値に応じて誤差拡散マトリックスを切り換える処理を行う場合もある。複数の画素については、階調値が近似あるいは同一であることが分かって

いれば、マトリックスや閾値を変更する処理を簡素なものとするのが可能となる。

#### 【0151】

以上、各種の実施例について説明してきたが、本発明は上記すべての実施例に限られるものではなく、その要旨を逸脱しない範囲において種々の態様で実施することができる。例えば、上述した機能を実現するソフトウェアプログラム（アプリケーションプログラム）を、通信回線を介してコンピュータシステムのメインメモリまたは外部記憶装置に供給し実行するものであってもよい。もちろん、CD-ROMやフレキシブルディスクに記憶されたソフトウェアプログラムを読み込んで実行するものであっても構わない。

#### 【0152】

また、上述した各種実施例では、ハーフトーン処理を含む画像データ変換処理はコンピュータ内で実行されるものとして説明したが、画像データ変換処理の一部あるいは全部をプリンタ側、あるいは専用の画像処理装置を用いて実行するものであっても構わない。

#### 【0153】

更には、画像表示装置は、必ずしも印刷媒体上にインクドットを形成して画像を印刷する印刷装置に限定されるものではなく、例えば、液晶表示画面上で輝点を適切な密度で分散させることにより、階調が連続的に変化する画像を表現する液晶表示装置であっても構わない。

#### 【図面の簡単な説明】

【図1】 本発明の概要を例示した印刷システムの概略構成図である。

【図2】 本実施例の画像処理装置としてのコンピュータの構成を示す説明図である。

【図3】 本実施例の画像表示装置としてのプリンタの概略構成図である。

【図4】 本実施例の画像処理装置で行われる画像処理の流れを示すフローチャートである。

【図5】 第1実施例の色変換・圧縮処理の流れを示すフローチャートである。



【図 6】 複数個の画素をまとめて第 1 のブロックを設定している様子を示す説明図である。

【図 7】 色変換のために参照される色変換テーブルを概念的に示した説明図である。

【図 8】 ブロック内で所定画素間の階調差を算出することによりエッジの有無を検出する様子を概念的に示した説明図である。

【図 9】 エッジの有無に応じてブロック内の画像データを圧縮し、メモリに記憶する様子を概念的に示した説明図である。

【図 10】 ブロック単位で行うハーフトーン処理の流れを示すフローチャートである。

【図 11】 ブロック単位で行うハーフトーン処理において設定される第 2 のブロックを例示した説明図である。

【図 12】 ハーフトーン処理において処理したブロックで発生した誤差を周辺の画素に拡散する様子を概念的に示した説明図である。

【図 13】 ブロック全体で生じた階調誤差を周辺の画素に拡散する際の拡散比率を例示した説明図である。

【図 14】 ブロック単位で行うハーフトーン処理中でブロック内の各画素について 2 値化を行う処理の流れを示したフローチャートである。

【図 15】 ブロック単位で行うハーフトーン処理中でブロック内の各画素について 2 値化を行う様子を概念的に示した説明図である。

【図 16】 第 2 実施例の色変換・圧縮処理の流れを示すフローチャートである。

【図 17】 第 3 実施例の色変換・圧縮処理の流れを示すフローチャートである。

【図 18】 ブロック内でエッジを検出しながら圧縮を行う変形例を概念的に示した説明図である。

【図 19】 エッジの状態に応じてブロックをサブブロックに分割しながら画像データを圧縮する変形例について示した説明図である。

【符号の説明】

10...コンピュータ  
12...プリンタドライバ  
20...カラープリンタ  
100...コンピュータ  
102...CPU  
104...ROM  
106...RAM  
108...周辺機器インターフェースP-I/F  
109...ディスクコントローラDDC  
110...ネットワークインターフェースカードNIC  
112...ビデオインターフェースV-I/F  
114...CRT  
116...バス  
118...ハードディスク  
120...デジタルカメラ  
122...カラーキャナ  
124...フレキシブルディスク  
126...コンパクトディスク  
200...カラープリンタ  
230...キャリッジモータ  
235...紙送りモータ  
236...プラテン  
240...キャリッジ  
241...印字ヘッド  
242...インクカートリッジ  
243...インクカートリッジ  
244...インク吐出用ヘッド  
260...制御回路  
261...CPU

2 6 2 ... R O M

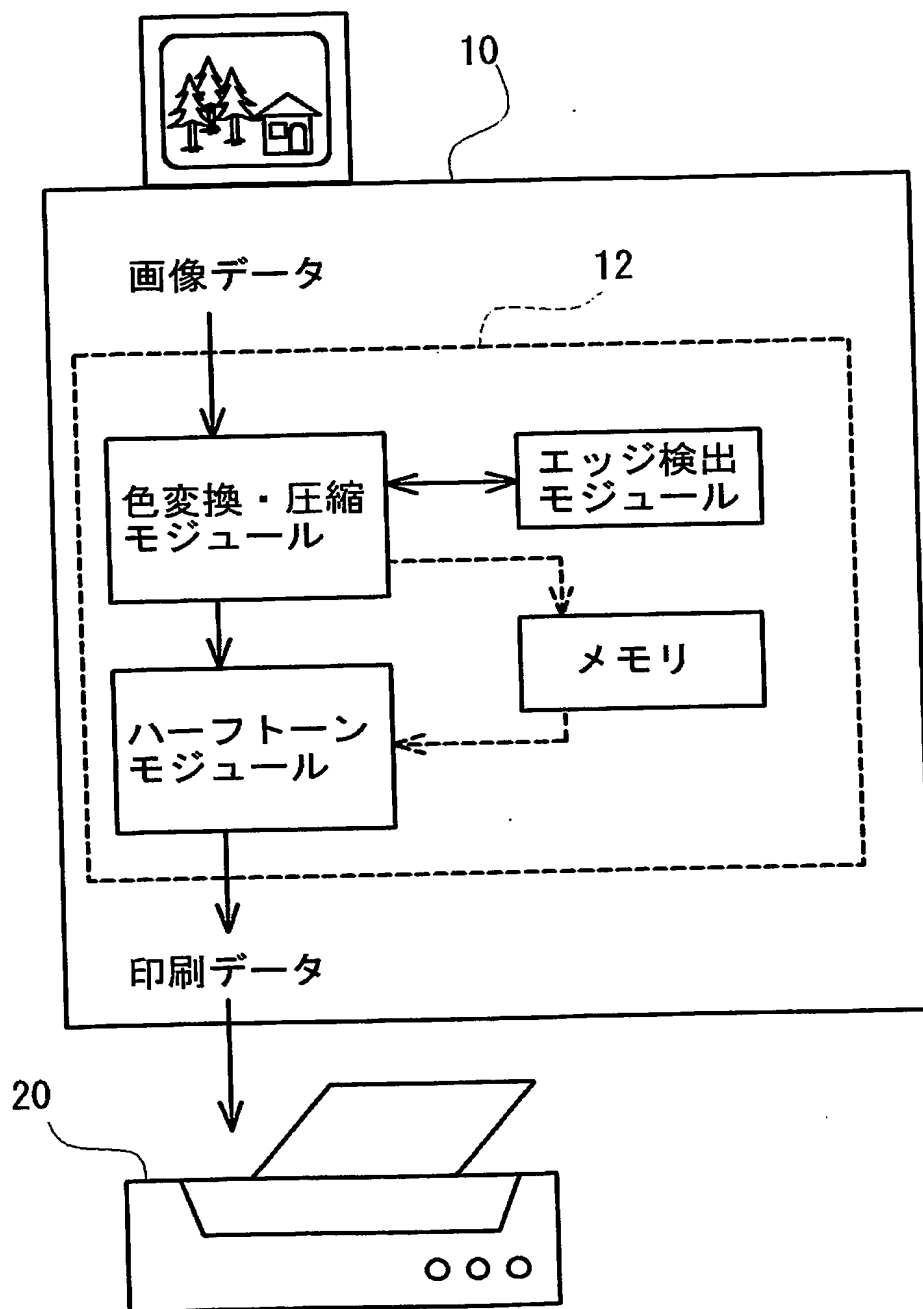
2 6 3 ... R A M

3 0 0 ... 通信回線

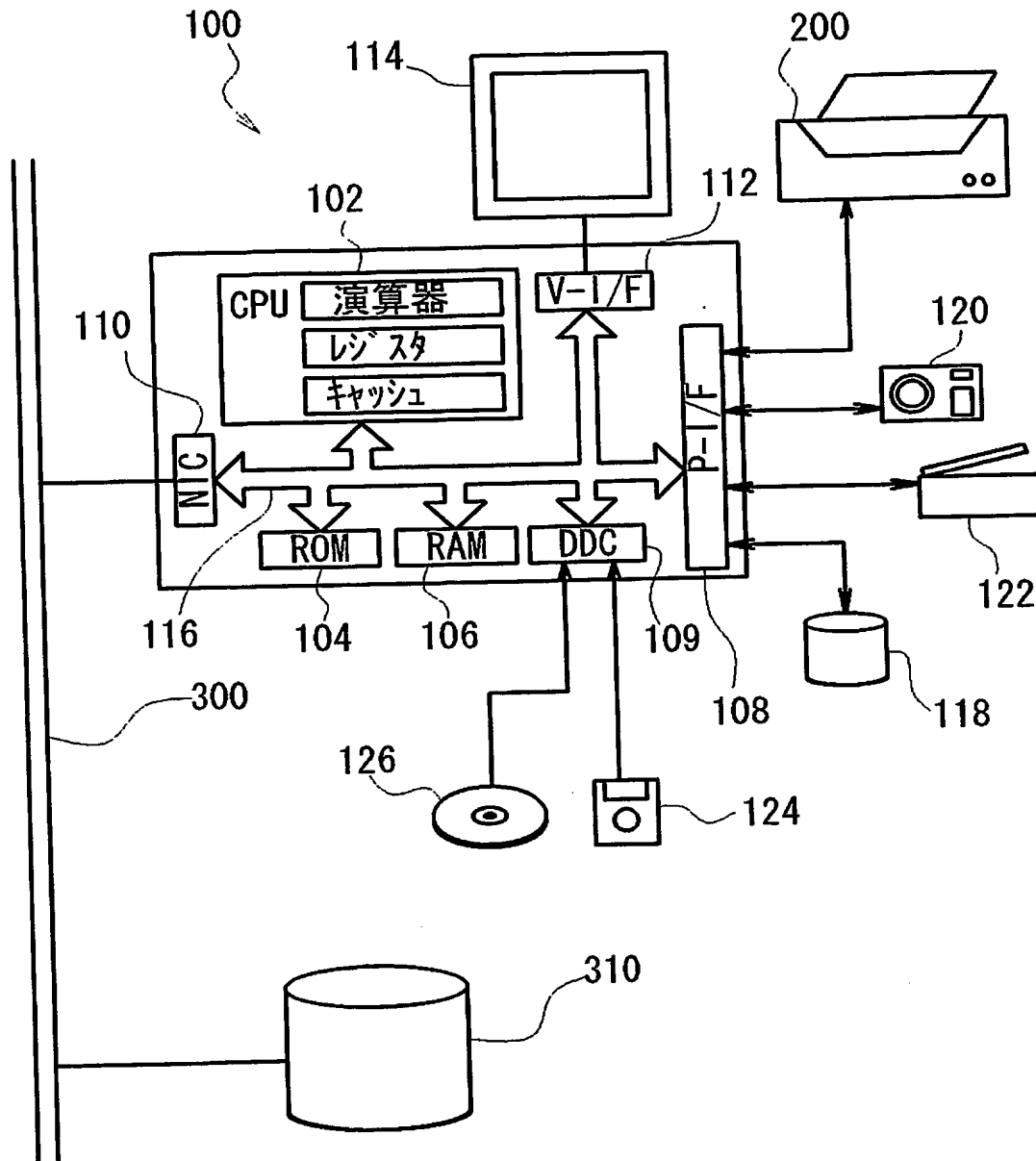
3 1 0 ... 記憶装置

【書類名】 図面

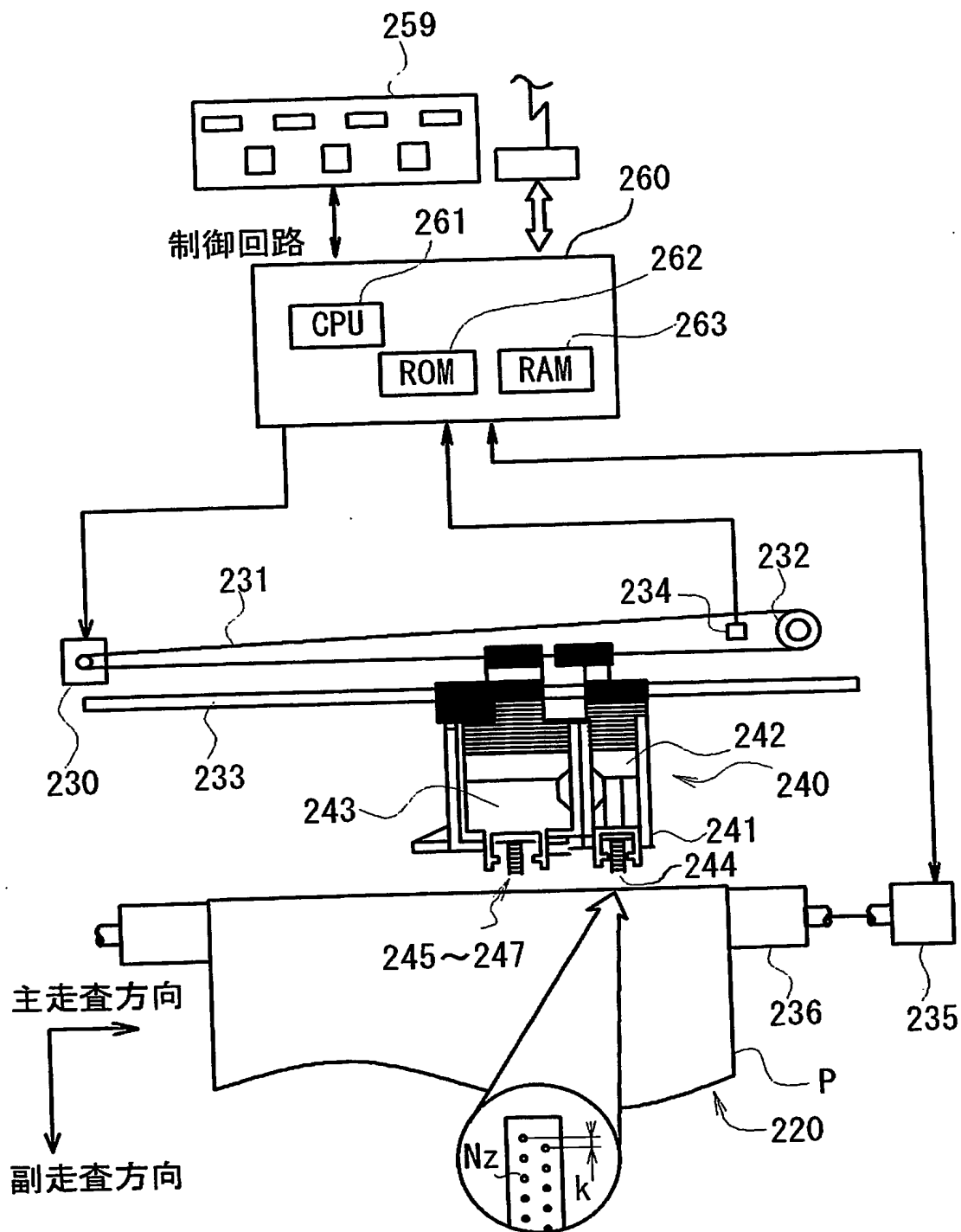
【図 1】



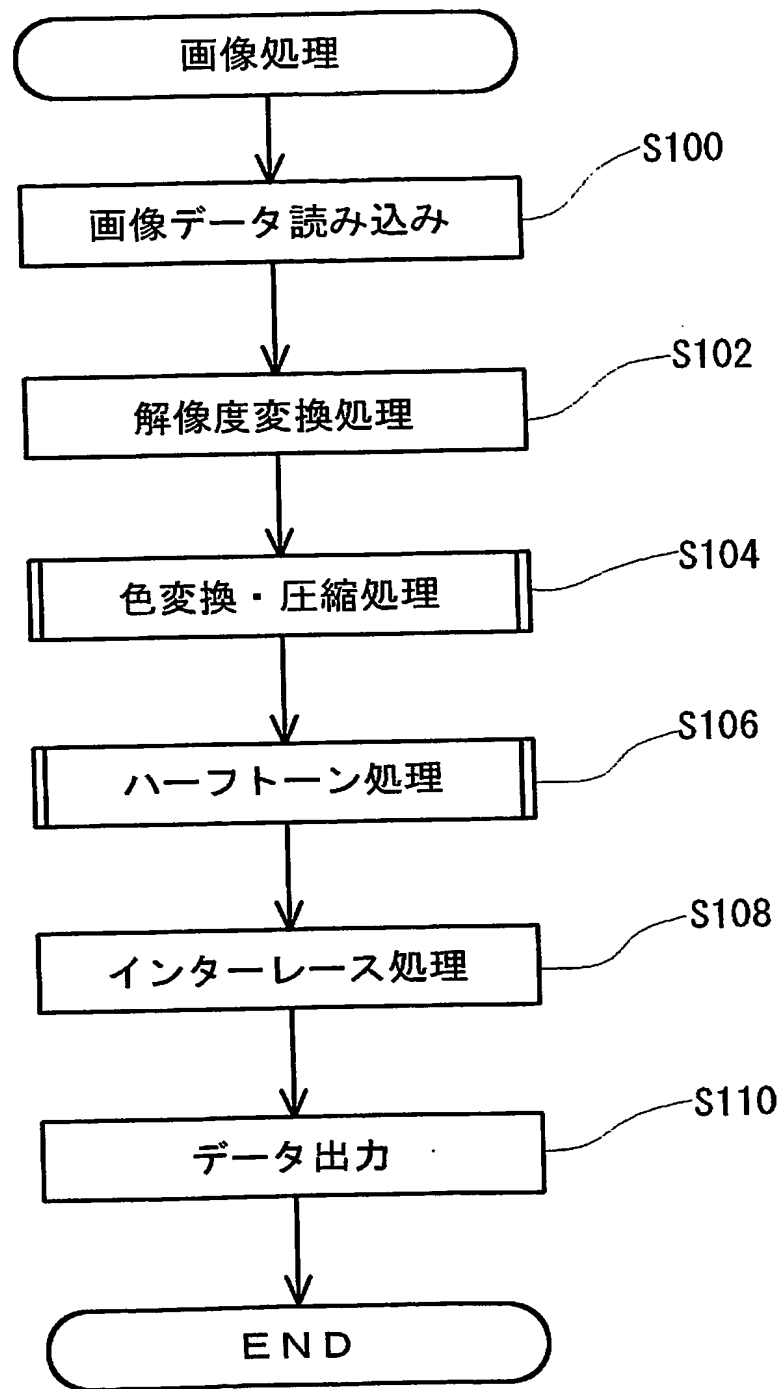
【図 2】



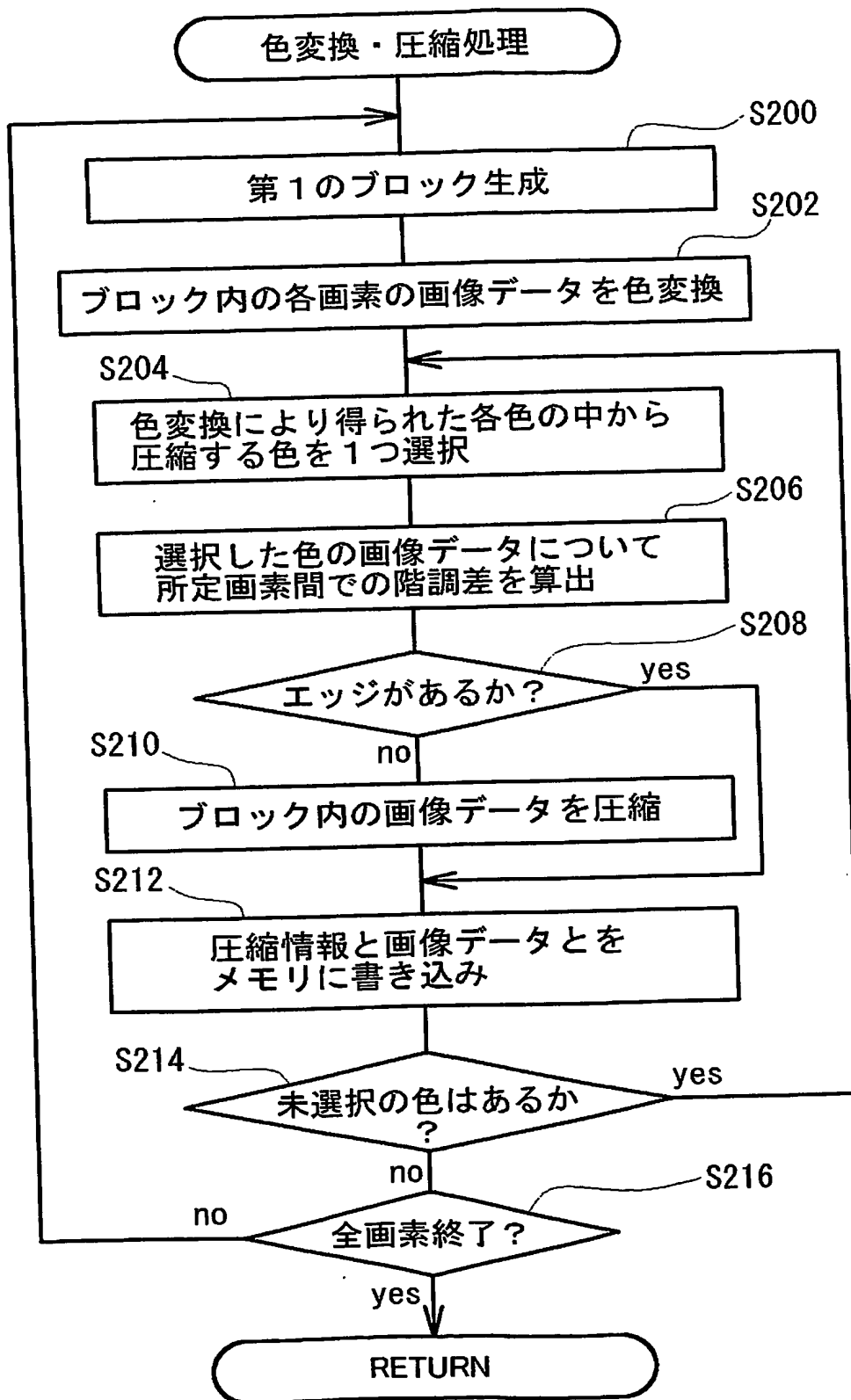
【図 3】



【図 4】

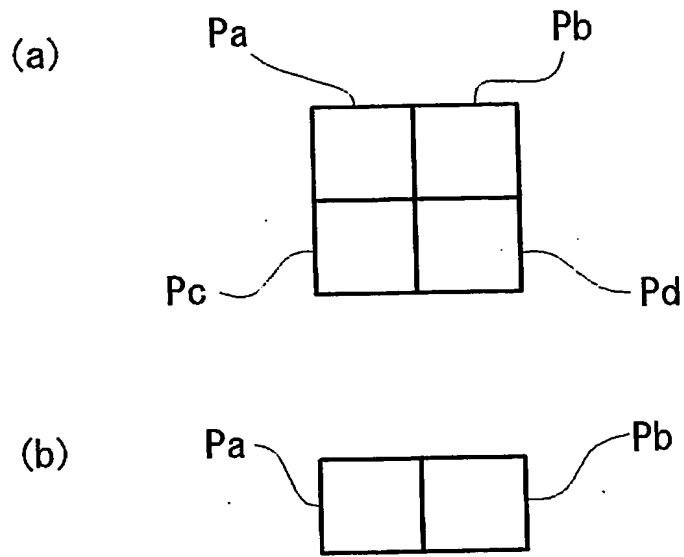


【図 5】

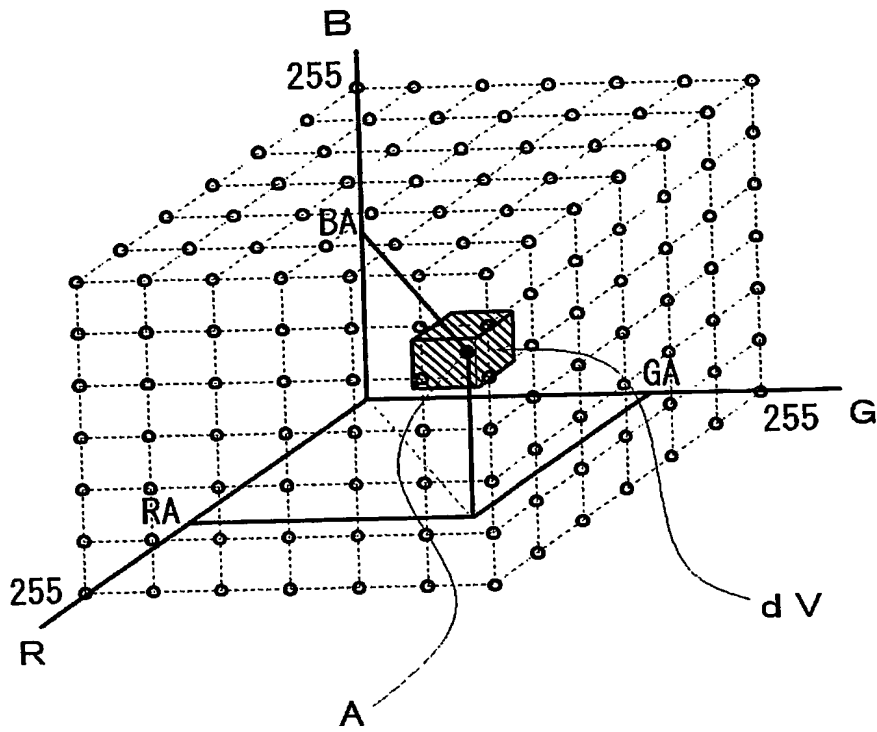




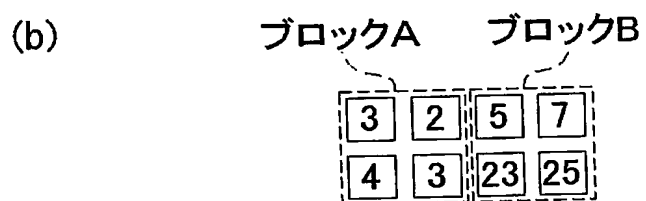
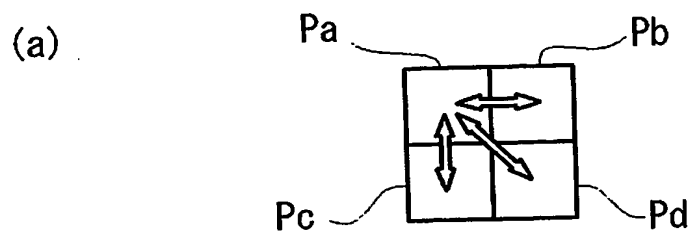
【図 6】



【図 7】

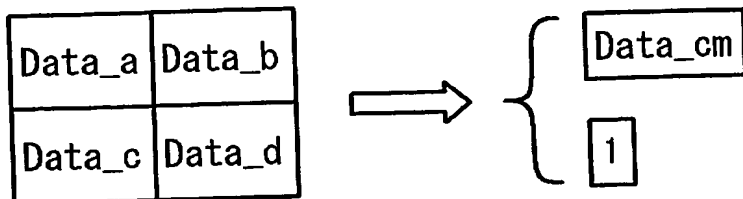


【図 8】

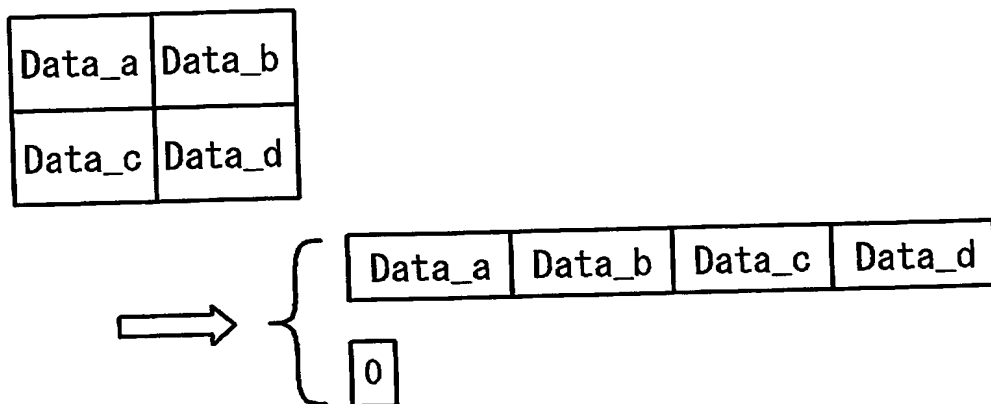


【図 9】

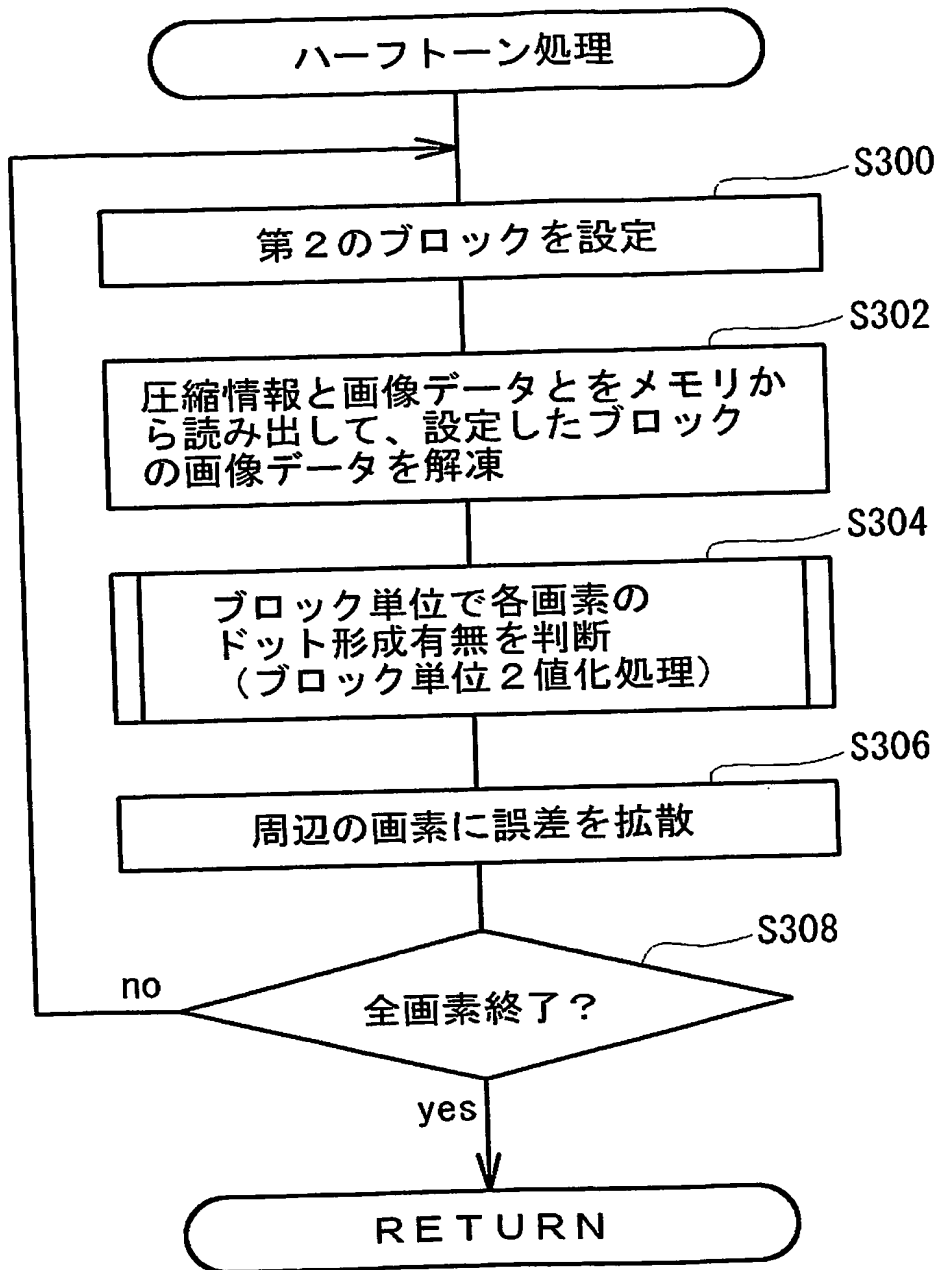
(a)



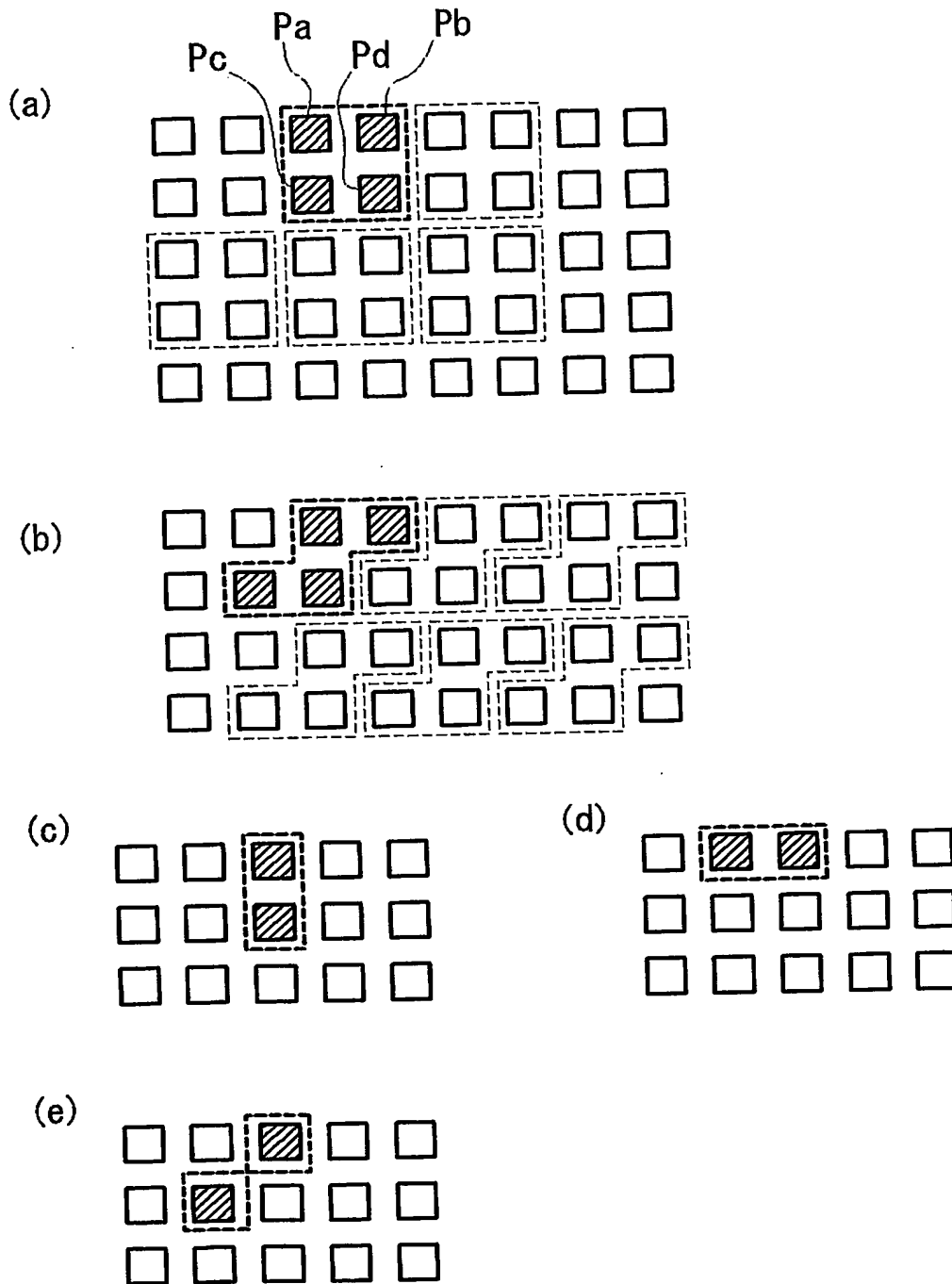
(b)



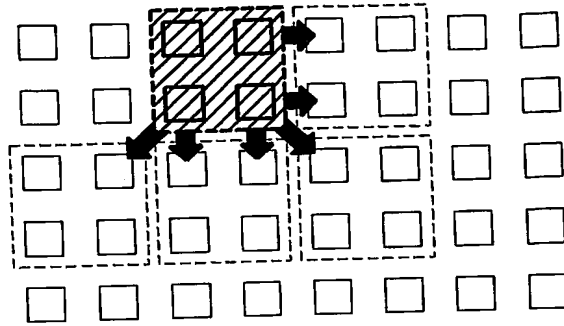
【図 10】



【図 11】

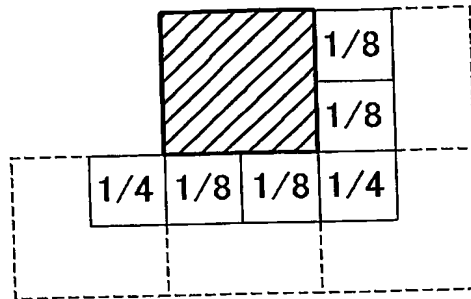


【図 12】

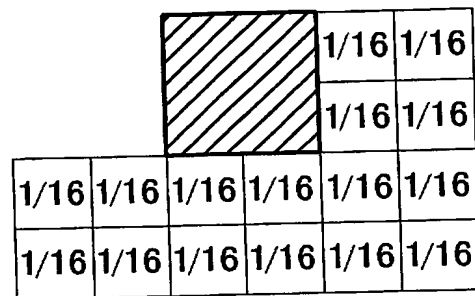


【圖 13】

(a)

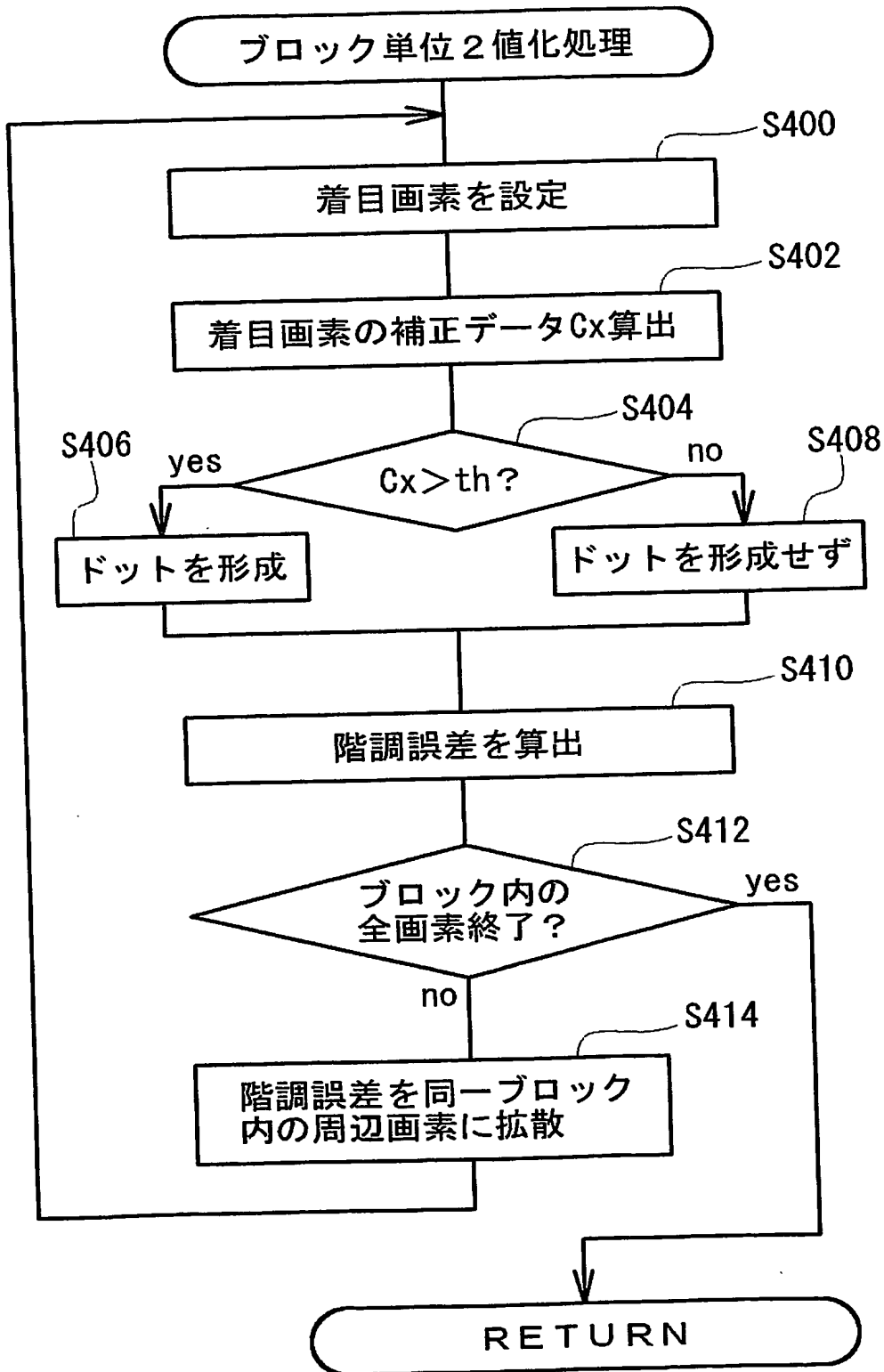


(b) .

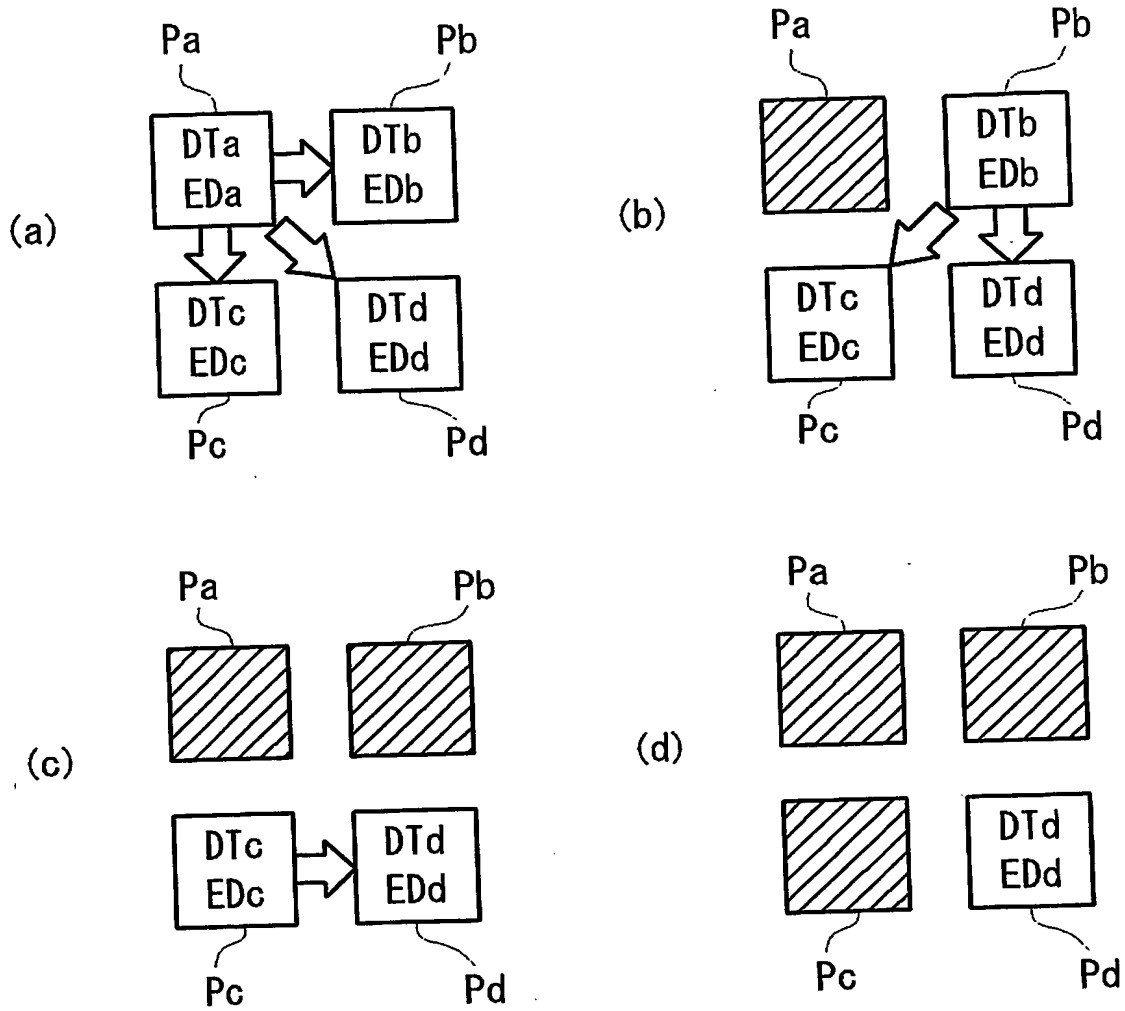




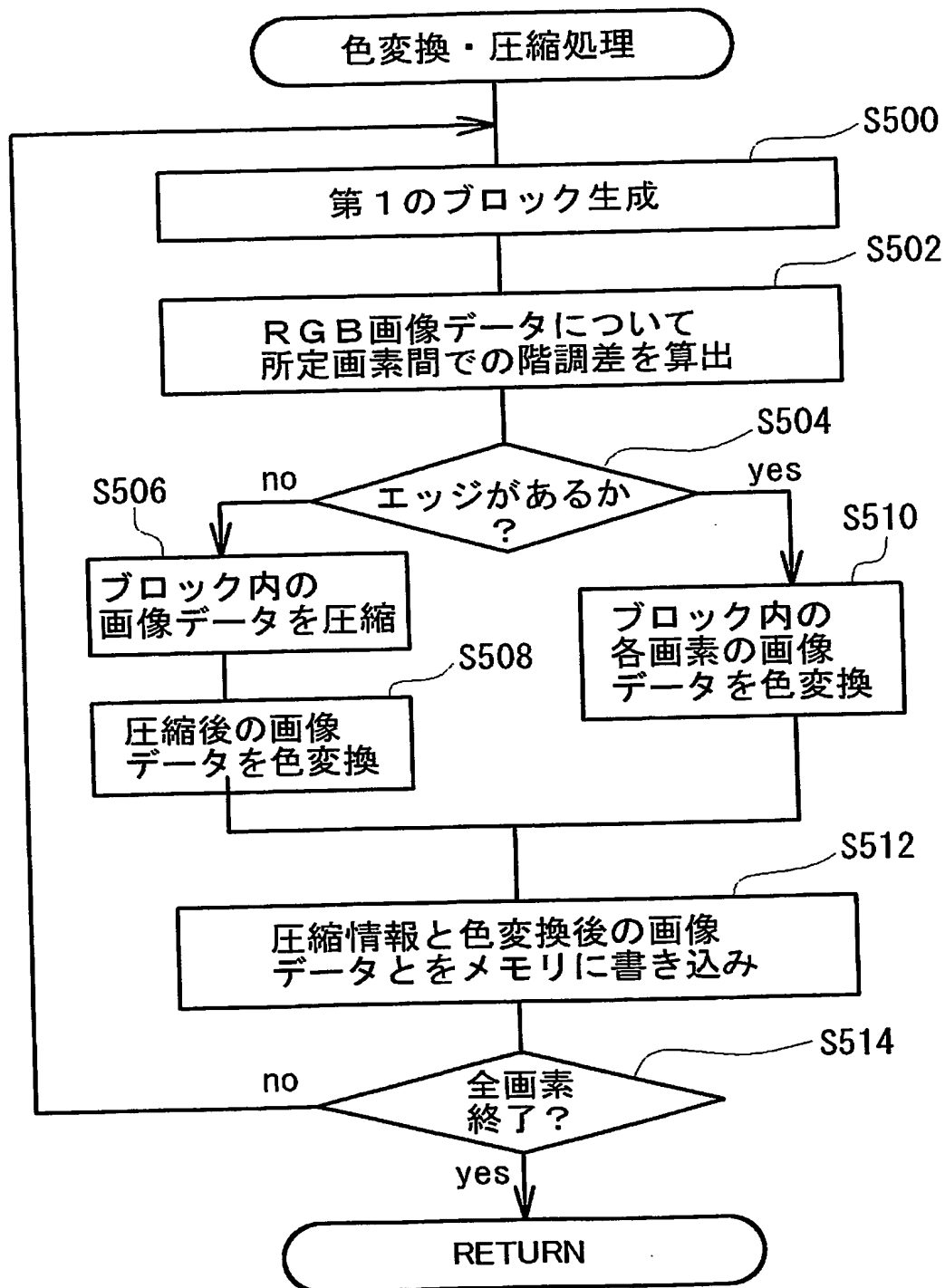
【図 14】



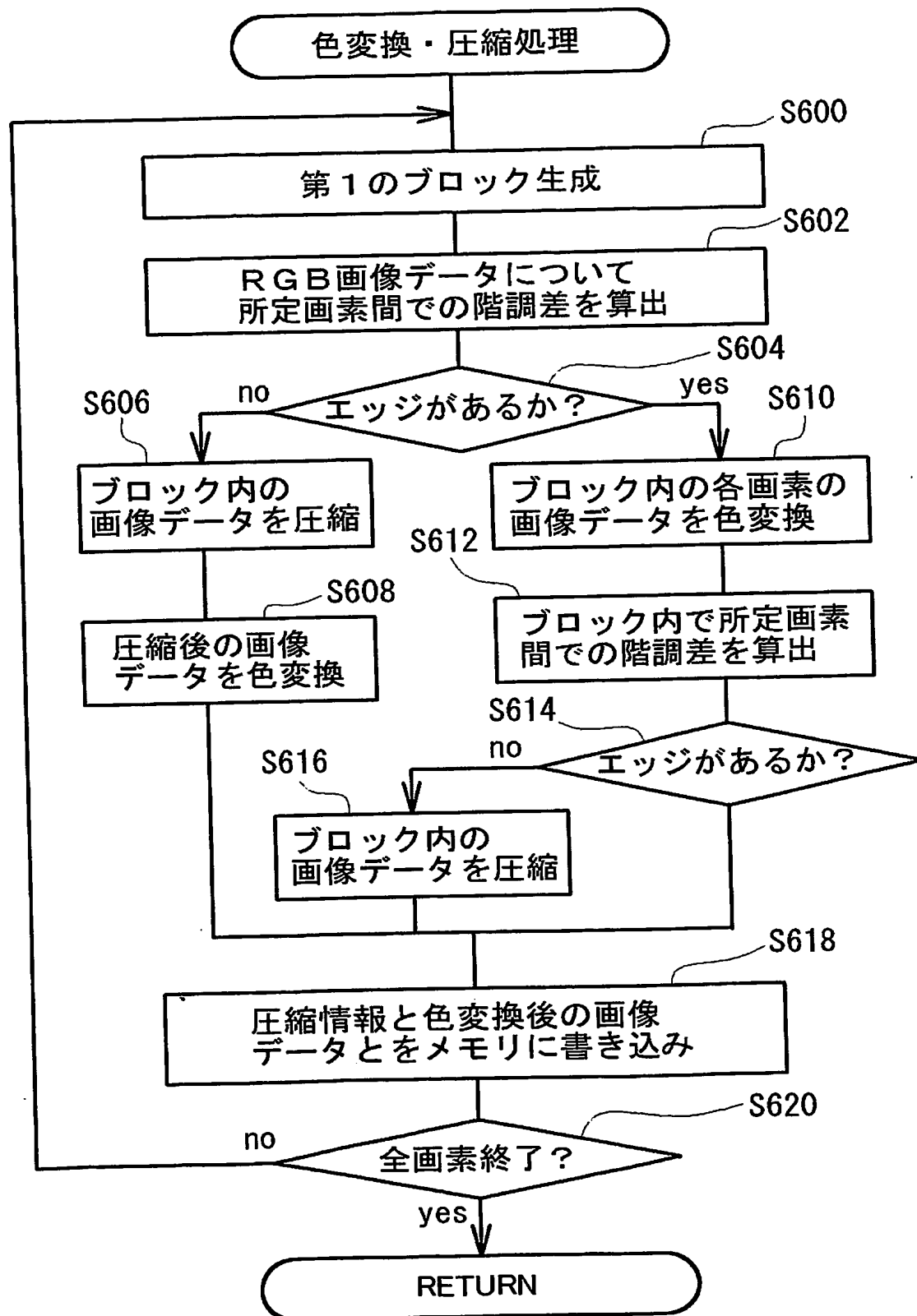
【図 15】



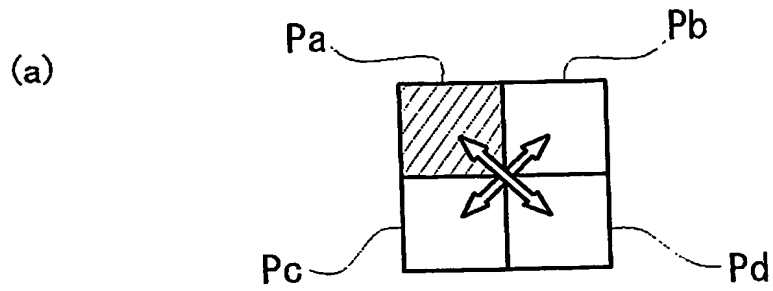
【図16】



【図 17】



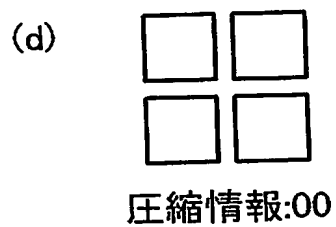
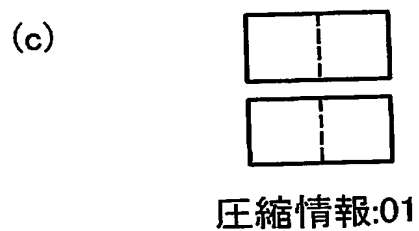
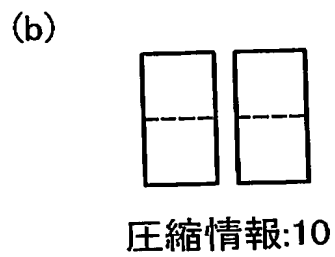
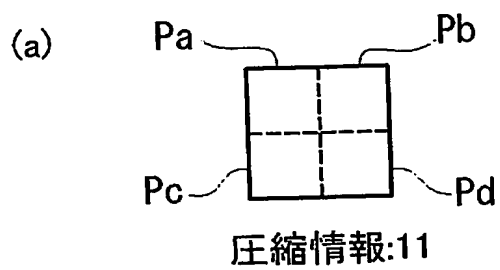
【図 18】



(b)

3	26
25	3

【図 19】



【書類名】 要約書

【要約】

【課題】 カラー画像データをドット形成有無により表現されたデータに迅速に変換する。

【解決手段】 隣接する複数の画素を第1種のブロックとしてまとめ、誤差拡散法を適用して行うドット形成有無の判断を、ブロック単位で行う。かかる判断に用いられる色変換後の画像データは、ブロック内でのエッジの有無に応じて圧縮された状態で一旦メモリに記憶しておき、必要に応じてメモリから読み出しながらドット形成の有無を判断する。こうすれば、ブロック単位で行うドット形成有無の判断が複雑化することを避けつつ、迅速に判断することができる。また、画像データはエッジの有無に応じて圧縮されているので、メモリに対して迅速に読み書きし、更に迅速に解凍することができる。このため、カラー画像データを色変換して、ドットの形成有無により表現されたデータに変換する処理を、処理内容を複雑化させることなく、迅速に行うことが可能となる。

【選択図】 図1

特願 2003-076071

出 願 人 履 歴 情 報

識別番号

[000002369]

1. 変更年月日

[変更理由]

住 所

氏 名

1990年 8月20日

新規登録

東京都新宿区西新宿2丁目4番1号

セイコーエプソン株式会社



This Page is inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record

## BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ BLACK BORDERS

☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES

☒ FADED TEXT OR DRAWING

☒ BLURED OR ILLEGIBLE TEXT OR DRAWING

☐ SKEWED/SLANTED IMAGES

☐ COLORED OR BLACK AND WHITE PHOTOGRAPHS

☐ GRAY SCALE DOCUMENTS

☐ LINES OR MARKS ON ORIGINAL DOCUMENT

☐ REPERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images  
problems checked, please do not report the  
problems to the IFW Image Problem Mailbox**